

Oliver Schütze  
Carlos A. Coello Coello  
Alexandru-Adrian Tantar  
Emilia Tantar  
Pascal Bouvry  
Pierre Del Moral  
Pierrick Legrand (Eds.)

# **EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II**

## **Editor-in-Chief**

Prof. Janusz Kacprzyk  
Systems Research Institute  
Polish Academy of Sciences  
ul. Newelska 6  
01-447 Warsaw  
Poland  
E-mail: kacprzyk@ibspan.waw.pl

Oliver Schütze, Carlos A. Coello Coello,  
Alexandru-Adrian Tantar, Emilia Tantar,  
Pascal Bouvry, Pierre Del Moral,  
and Pierrick Legrand (Eds.)

---

EVOLVE – A Bridge  
between Probability,  
Set Oriented Numerics,  
and Evolutionary  
Computation II

 Springer

*Editors*

Dr. Oliver Schütze  
CINVESTAV-IPN  
Depto. de Ingeniería Eléctrica  
Sección de Computación  
Mexico City  
Mexico

Dr. Carlos A. Coello Coello  
CINVESTAV-IPN  
Depto. de Computación  
Mexico City  
Mexico

Dr. Alexandru-Adrian Tantar  
University of Luxembourg  
Computer Science and Communications  
Research Unit  
Luxembourg

Dr. Emilia Tantar  
University of Luxembourg  
Computer Science and Communications  
Research Unit  
Luxembourg

Dr. Pascal Bouvry  
Faculty of Sciences, Technology  
and Communication  
University of Luxembourg  
Computer Science and Communication  
Group  
Luxembourg

Dr. Pierre Del Moral  
Bordeaux Mathematical Institute  
Université Bordeaux I  
Talence cedex  
France

Dr. Pierrick Legrand  
Université Bordeaux2, Bâtiment Leyteire  
URF Sciences et Modelisation  
Bordeaux  
France

ISSN 2194-5357  
ISBN 978-3-642-31518-3  
DOI 10.1007/978-3-642-31519-0  
Springer Heidelberg New York Dordrecht London

e-ISSN 2194-5365  
e-ISBN 978-3-642-31519-0

Library of Congress Control Number: 2012940858

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

# Preface

The massive use and large applicability spectrum of evolutionary algorithms for real-life applications determined the need of establishing solid theoretical grounds. Only to offer one example, one may consider mathematical objects that are sometimes difficult and/or costly to calculate. At the same time, acknowledged new results show that evolutionary computation can provide in some cases good and fast estimators of such quantities. Similarly, the handling of large quantities of data may require the use of distributed environments where the probability of failure and the stability of the algorithms may need to be addressed. What is more, common practice confirms in many cases that theory-based results have the advantage of ensuring performance guarantee factors for evolutionary algorithms in areas as diverse as optimization, bio-informatics or robotics.

The aim of the EVOLVE is to build a bridge between probability, statistics, set oriented numerics and evolutionary computing, as to identify new common and challenging research aspects. The conference is also intended to foster a growing interest for robust and efficient methods with a sound theoretical background. EVOLVE is intended to unify theory-inspired methods and cutting-edge techniques ensuring performance guarantee factors. By gathering researchers with different backgrounds, ranging from computer science to mathematics, statistics and physics, to name just a few, a unified view and vocabulary can emerge where the theoretical advancements may echo in different domains.

Summarizing, the EVOLVE focuses on challenging aspects arising at the passage from theory to new paradigms and aims to provide a unified view while raising questions related to reliability, performance guarantees and modeling.

This book contains the proceedings of EVOLVE 2012, organized as an international conference for the first time, after previous editions when it followed a workshop format. The EVOLVE series started in 2011 with an international workshop held at the Bourglinster Castle in Luxembourg, while in 2010, the originating event, Workshop on Evolutionary Algorithms - New Challenges in Theory and Practice, was organized in Bordeaux. The EVOLVE 2012 has been hosted by the Computer Science Department of the CINVESTAV-IPN, in Mexico City, Mexico.

This book consists of the accepted full-length papers that were submitted to the EVOLVE 2012 and that were peer-reviewed by an international program committee. For convenience of the reader we have divided the 32 papers into 8 main parts representing different research areas within the scope of the EVOLVE.

Part I consists of two invited papers coming from Keynote Speakers of the EVOLVE 2012. The first paper, by Jian-Qiao Sun, deals with the control of non-linear dynamic systems with the cell mapping method. The second paper, by Jose Blanchet et al., is about a method for estimating the quasi-stationary distribution of various particle processes.

In Part II, a group of papers is presented dealing with genetic programming. These contributions come from a special session organized by Leonardo Trujillo and Edgar Galvan.

In Part III, a collection of papers is given that contribute to the field of evolutionary multi-objective optimization. These papers come from a special session organized by Günter Rudolph and Heike Trautmann.

Part IV contains two papers dealing with combinatorial optimization. The first one presents a hyperheuristic approach for guiding enumeration in constraint solving, and the second one proposes a simulated annealing implementation for finding near-optimal solutions for the Maximum Parsimony problem.

Part V contains one contribution to the field of probabilistic modeling and optimization for emerging networks, coming from a special session organized by Jianguo Ding and Xinhui Wang.

In Part VI, a selection of research works on hybrid probabilistic models for real parameter optimization and their applications are presented. These papers come from a special session organized by Arturo Hernández-Aguirre.

In Part VII, a group of papers contributing to the field of evolutionary computation for vision, graphics, and robotics are presented. These works come from a special session organized by Gustavo Olague and Humberto Sossa.

Finally, Part VIII presents a selection of contributed papers describing the application of bio-inspired metaheuristics to real problems of industrial and scientific research and development. These papers come from a special session organized by Andrew Lewis and Marcus Randall.

We would like to express our gratitude to all the invited speakers to accept our invitation and to give an outstanding presentation at the EVOLVE 2012. Further, we would like to thank the chairs of the special sessions, the members of the program committee, and the authors who have submitted a contribution to the event, which allowed to constitute this book and which allowed to make the EVOLVE 2012 a success. Finally, we gratefully thank the sponsors of the event and our institutions which helped us to realize our projects.

Mexico City, Luxembourg, and Bordeaux,  
August 2012

Oliver Schütze  
Carlos A. Coello Coello  
Alexandru-Adrian Tantar  
Emilia Tantar  
Pascal Bouvry  
Pierre del Moral  
Pierrick Legrand

# Organization

## Conference General Chairs

Oliver Schütze	CINVESTAV-IPN, Mexico
Alexandru-Adrian Tantar	University of Luxembourg, Luxembourg
Emilia Tantar	University of Luxembourg, Luxembourg
Pascal Bouvry	University of Luxembourg, Luxembourg
Pierre del Moral	INRIA Bordeaux-Sud Ouest, France
Pierrick Legrand	University of Bordeaux 2, France

## Advisory Board

Enrique Alba	University of Málaga, Spain
François Caron	INRIA Bordeaux Sud-Ouest, France
Frédéric Céro	INRIA Rennes Bretagne Atlantique, France
Carlos A. Coello Coello	CINVESTAV-IPN, México
Michael Dellnitz	University of Paderborn, Germany
Frédéric Guinand	University of Le Havre, France
Arnaud Guyader	Université Rennes 2, INRIA Rennes Bretagne Atlantique, France
Arturo Hernandez	CIMAT, México
Günter Rudolph	TU Dortmund University, Germany
Marc Schoenauer	INRIA Saclay - Ile-de-France, University Paris Sud, France
Franciszek Seredynski	Polish Academy of Sciences, Warsaw, Poland
El-Ghazali Talbi	Polytech' Lille, University of Lille 1, France
Marco Tomassini	University of Lausanne, Switzerland
Massimiliano Vasile	University of Strathclyde, Scotland

## Programm Committee

Nicola Beume	TU Dortmund University, Germany
Peter A.N. Bosman	CWI, The Netherlands
Jair Cervantes	UAEM-Texcoco, México

Edgar Chavez	University of Michoacana, México
Stephen Chen	York University, Canada
Francisco Chicano	University of Málaga, Spain
Christian Dominguez Medina	CIC-IPN, México
Liliana Cucu-Grosjean	Loria, France
Bernabe Dorronsoró	University of Luxembourg, Luxembourg
Michael Emmerich	Leiden University, The Netherlands
Edgar Galvan	Trinity College University, Ireland
Jesus Gonzalez Bernal	National Institute of Astrophysics, Optics and Electronics, México
Jeffrey Horn	Northern Michigan University, USA
Didier Keymeulen	Jet Propulsion Laboratory, USA
Joanna Kolodziej	University of Bielsko-Biala, Poland
Ricardo Landa Becerra	CINVESTAV-IPN, Mexico
Adriana Lara	IPN, Mexico
Andrew Lewis	Griffith University, Australia
Lili Liu	Northeastern University, China
Francisco Luna	University of Málaga, Spain
Gabriel Luque	University of Málaga, Spain
Evelyne Lutton	INRIA Saclay Ile-de-France, France
Luis Martí	Universidad Carlos III de Madrid, Spain
Jörn Mehnen	Cranfield University, UK
Nicolas Monmarché	University of Tours, France
James Montgomery	Swinburne University of Technology, Australia
Irene Moser	Swinburne University of Technology, Australia
Boris Naujoks	TU Dortmund University, Germany
Sergio Nesmachnov	Universidad de la República, Uruguay
Gustavo Olague	CICESE Research Center, México
Eduardo Rodriguez-Tello	CINVESTAV-IPN, Mexico
Gustavo Sanchez	Simon Bolivar University, Venezuela
Christoph Schommer	University of Luxembourg, Luxembourg
Antonio del Sol	University of Luxembourg, Luxembourg
Juan Humberto Sossa Azuela	CIC-IPN, México
Kiyoshi Tanaka	Shinshu University, Japan
Gregorio Toscano-Pulido	CINVESTAV-IPN, México
Heike Trautmann	TU Dortmund University, Germany
Leonardo Trujillo	CICESE Research Center, Mexico
Alan Reynolds	Heriot-Watt University, Edinburgh, Scotland
Hiroyuki Sato	Shinshu University, Japan
Ponnuthurai Suganthan	Nanyang Technological University, Singapore
Simon Wessing	TU Dortmund University, Germany
Fatos Xhafa	Universitat Politecnica de Catalunya, Spain



**Local Organizing Committee**

Oliver Schütze	CINVESTAV-IPN, México
Felipa Rosas López	CINVESTAV-IPN, México
Sofía Reza Cruz	CINVESTAV-IPN, México
Erika Berenice Ríos Hernández	CINVESTAV-IPN, México
Santiago Domínguez Domínguez	CINVESTAV-IPN, México
José Luis Flores Garcilazo	CINVESTAV-IPN, México
Christian Dominguez Medina	CIC-IPN, México

# Contents

Organization .....	VI
--------------------	----

## Part I Cell Mapping and Quasi-stationary Distributions

<b>Control of Nonlinear Dynamic Systems with the Cell Mapping Method</b> .....	3
--	---

*Jian-Qiao Sun*

1 Introduction .....	3
2 Optimal Control .....	4
3 Cell Mapping Methods .....	6
3.1 Control Application .....	6
4 Optimal Control of Competing Species .....	7
4.1 Numerical Examples of Optimal Control .....	9
5 Conclusions .....	16
References .....	17

<b>Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions</b> .....	19
---	----

*Jose Blanchet, Peter Glynn, Shuheng Zheng*

1 Introduction .....	19
1.1 Related Literature .....	20
2 Background and Motivation .....	21
2.1 Contact Process .....	21
2.2 Quasi-stationary Distribution .....	22
2.3 Physicist's Heuristic .....	23
3 Stochastic Approximation Analysis of the Algorithm .....	24
3.1 Formal Description of the Algorithm .....	24
3.2 Sketch of Proof of Convergence .....	26
4 Variations on the Existing Algorithm with Improved Rate of Convergence .....	27
4.1 Counter Example to CLT .....	27
4.2 The Parallel Algorithm .....	28

5	Continuous-Time Markov Chains . . . . .	30
5.1	Formulation and Convergence . . . . .	30
5.2	Rate of Convergence . . . . .	31
5.3	Uniformization . . . . .	31
6	Numerical Experiments . . . . .	31
6.1	Loopy Markov Chain . . . . .	31
6.2	Contact Process on Complete Graph . . . . .	34
7	Discussion and Conclusion . . . . .	36
	References . . . . .	36

## Part II Genetic Programming

### Locality in Continuous Fitness-Valued Cases and Genetic Programming

<b>Difficulty</b> . . . . .	<b>41</b>
-----------------------------	-----------

*Edgar Galvan, Leonardo Trujillo, James McDermott, Ahmed Kattan*

1	Introduction . . . . .	42
2	Locality . . . . .	43
2.1	Extending the Definition of Locality to the Continuous Valued-Fitness . . . . .	44
3	Related Work . . . . .	45
3.1	Sampling the Fitness Landscape . . . . .	45
4	Experimental Setup . . . . .	47
4.1	Benchmark Problems . . . . .	47
4.2	Uniform Genetic Programming . . . . .	47
4.3	Setting Bounds . . . . .	48
4.4	Evolutionary Runs . . . . .	48
4.5	Sampling and Measuring Locality . . . . .	49
5	Results and Discussion . . . . .	49
5.1	Quantitative Locality Measures and Performance . . . . .	49
5.2	Definitions of Locality and Limitations . . . . .	54
6	Conclusions . . . . .	54
	References . . . . .	55

### Analysis and Classification of Epilepsy Stages with Genetic

<b>Programming</b> . . . . .	<b>57</b>
------------------------------	-----------

*Arturo Sotelo, Enrique Guijarro, Leonardo Trujillo, Luis Coria,*

*Yuliana Martínez*

1	Introduction . . . . .	57
2	Epilepsy Signals . . . . .	59
3	Experimental Data . . . . .	60
3.1	Signal Recording . . . . .	61
4	Problem Statement . . . . .	63
4.1	Proposal . . . . .	63
5	Experiments and Results . . . . .	65
6	Summary and Conclusions . . . . .	68
	References . . . . .	69

<b>Disparity Map Estimation by Combining Cost Volume Measures Using Genetic Programming</b> .....	<b>71</b>
<i>Enrique Naredo, Enrique Dunn, Leonardo Trujillo</i>	
1 Introduction .....	72
2 Stereo Computer Vision .....	73
2.1 Disparity Map .....	73
2.2 Correspondence Algorithms .....	74
3 Proposal .....	76
3.1 Problem Statement .....	76
3.2 Proposed Solution .....	76
4 Experimental Configuration and Results .....	76
4.1 Dataset .....	76
4.2 GP Search .....	77
4.3 Experimental Setup .....	78
4.4 Experimental Results .....	79
5 Conclusions and Future Work .....	84
References .....	85

### Part III Evolutionary Multi-objective Optimization

<b>Finding Evenly Spaced Pareto Fronts for Three-Objective Optimization Problems</b> .....	<b>89</b>
<i>Heike Trautmann, Günter Rudolph, Christian Dominguez-Medina, Oliver Schütze</i>	
1 Introduction .....	89
2 Background .....	91
3 Methods .....	92
3.1 Base EMOA .....	92
3.2 Tools .....	93
3.3 Specialized EMOA for Evenly Spaced Pareto Fronts .....	95
4 Experiments .....	96
4.1 Setup .....	98
4.2 Results .....	99
5 Conclusions .....	103
References .....	103

<b>Software Requirements Optimization Using Multi-Objective Quantum-Inspired Hybrid Differential Evolution</b> .....	<b>107</b>
<i>A. Charan Kumari, K. Srinivas, M.P. Gupta</i>	
1 Introduction .....	108
2 Multi-Objective Optimization .....	109
2.1 Multi-Objective Optimization Problem (MOP) .....	109
2.2 Pareto Dominance .....	110
2.3 Pareto Optimality .....	110
2.4 Pareto Optimal Set .....	110
2.5 Pareto Front .....	110

3	Multi-Objective Next Release Problem .....	111
4	Multi-Objective Quantum-Inspired Hybrid Differential Evolution .....	112
4.1	Representation .....	112
4.2	Mutation Operator .....	113
4.3	Crossover Operator .....	114
4.4	Selection .....	114
5	Experiments .....	114
5.1	Test Problems .....	114
5.2	Performance Measures .....	115
5.3	Experimental Setup .....	116
5.4	Methodology .....	117
6	Results and Analysis .....	117
7	Conclusions .....	119
	References .....	120

**Evolving a Pareto Front for an Optimal Bi-objective Robust Interception Problem with Imperfect Information .....** 121

*Gideon Avigad, Erella Eisenstadt, Valery Y. Glizer*

1	Introduction and Problem Formulation .....	121
2	Background .....	123
3	Methodology .....	124
3.1	Pursuer's Control in a No-Game Situation .....	124
3.2	Pursuit with Uncertainties: The Game Situation .....	126
3.3	The MOE Algorithm .....	131
3.4	Simulations for the Game at Hand .....	132
4	Summary and Conclusions .....	134
	References .....	134

**PSA – A New Scalable Space Partition Based Selection Algorithm for MOEAs .....** 137

*Shaul Salomon, Gideon Avigad, Alex Goldvard, Oliver Schütze*

1	Introduction .....	137
2	PSA – Part and Select Algorithm .....	139
2.1	Partitioning a Set .....	139
2.2	Selection of a Representative Subset .....	141
2.3	Complexity Analysis of the PSA .....	142
3	Integration of PSA into NSGA-II .....	144
4	Comparison of NSGA2-PSA with NSGA-II .....	144
4.1	Test Problems .....	145
4.2	Genetic Settings .....	145
4.3	Performance Metrics .....	145
4.4	Simulation Results .....	149
5	Conclusions and Future Work .....	149
	References .....	150

<b>The Gradient Free Directed Search Method as Local Search within Multi-Objective Evolutionary Algorithms</b> .....	<b>153</b>
<i>Adriana Lara, Sergio Alvarado, Shaul Salomon, Gideon Avigad, Carlos A. Coello Coello, Oliver Schütze</i>	
1 Introduction .....	153
2 Background .....	154
3 Gradient Free Directed Search .....	156
4 Integration of DDS into MOEA/D .....	160
5 Numerical Results .....	162
5.1 Comparison MOEA/D and MOEA/D/DDS .....	162
5.2 A Control Problem .....	164
6 Conclusions .....	166
References .....	166

## Part IV Combinatorial Optimization

<b>A Hyperheuristic Approach for Guiding Enumeration in Constraint Solving</b> .....	<b>171</b>
<i>Broderick Crawford, Carlos Castro, Eric Monfroy, Ricardo Soto, Wenceslao Palma, Fernando Paredes</i>	
1 Introduction .....	171
2 A Hyperheuristic Approach for Dynamic Selection of Enumeration Strategies .....	173
2.1 The Hyperheuristic Approach .....	174
2.2 Measuring the Solving Process: Indicators .....	175
2.3 Choice Function .....	176
2.4 Choice Function Tuning with a Multilevel Structure .....	176
3 Experimental Evaluation .....	178
3.1 Results .....	179
4 Conclusions .....	186
References .....	187

<b>Maximum Parsimony Phylogenetic Inference Using Simulated Annealing</b> .....	<b>189</b>
<i>Jean-Michel Richer, Eduardo Rodriguez-Tello, Karla E. Vazquez-Ortiz</i>	
1 Introduction .....	189
2 An Improved Implementation of a Simulated Annealing Algorithm .....	192
2.1 Internal Representation and Search Space .....	193
2.2 Evaluation Function .....	193
2.3 Initial Solution .....	193
2.4 Neighborhood Functions .....	194
2.5 Cooling Schedule .....	194
2.6 Stop Condition .....	195
3 Computational Experiments .....	196
3.1 Benchmark Instances and Performance Assessment .....	196
3.2 Components and Parameters Tuning .....	196

3.3	Comparing SAMPARS with an Existing SA Implementation .....	199
3.4	Comparing SAMPARS with the State-of-the-Art Procedures .....	200
4	Conclusions .....	201
	References .....	202

## Part V Probabilistic Modeling and Optimization for Emerging Networks

<b>A Bayesian Network Based Critical Infrastructure Risk Model .....</b>		<b>207</b>
<i>Thomas Schaberreiter, Pascal Bouvry, Juha Röning, Djamel Khadraoui</i>		
1	Introduction .....	207
2	Related Work .....	208
3	Introduction to Bayesian Networks .....	210
4	BN Based Critical Infrastructure Security Model .....	211
4.1	Structure for Bayesian Network .....	212
4.2	Conditional Probability Tables .....	213
4.3	Risk Prediction .....	215
4.4	Interdependencies - Directed Cycles in Bayesian Networks .....	215
5	Conclusions and Future Work .....	217
	References .....	218

## Part VI Hybrid Probabilistic Models for Real Parameter Optimization and Their Applications

<b>Adequate Variance Maintenance in a Normal EDA via the Potential-Selection Method .....</b>		<b>221</b>
<i>P.S. Ivvan Valdez, Arturo Hernández-Aguirre, and Salvador Botello</i>		
1	Introduction .....	221
1.1	Sampling Intensively the Most Promising Regions .....	222
1.2	Preserving and/or Computing Adequate Variance Parameters in Order to Maintain the Exploration Capacity .....	223
2	Selection Method .....	224
3	Potential Selection Based EDA .....	226
4	Experiments and Results Discussion .....	227
4.1	Comparison with MIDEA, MBOA, and EDA with vbICA-MM .....	230
4.2	Comparison with Adaptive Variance Scaling (Grahl et al.) .....	230
4.3	Comparison with Adaptive Variance Scaling Schemes (Yungpeng et al.) .....	232
5	Conclusions .....	233
	References .....	234

<b>Linkage Learning Using Graphical Markov Model Structure: An Experimental Study</b> .....	<b>237</b>
<i>Eunice Esther Ponce-de-Leon-Senti, Elva Diaz-Diaz</i>	
1 Introduction .....	237
2 Definitions and Concepts .....	238
3 The Adaptive Extended Tree Cliqued - EDA (AETC-EDA) .....	242
3.1 Adaptive Extended Tree Cliqued – EDA (AETC–EDA) Pseudocode .....	244
4 Test Functions .....	244
5 Experimental Results .....	246
6 Discussion and Conclusions .....	248
References .....	248
<b>A Comparison Study of PSO Neighborhoods</b> .....	<b>251</b>
<i>Angel Eduardo Muñoz Zavala</i>	
1 Introduction .....	251
2 Neighborhood Topologies .....	253
2.1 Ring Topology .....	254
2.2 Von Neumann Topology .....	254
2.3 Singly-Linked Ring Topology .....	255
3 Experiments .....	256
3.1 Test I .....	258
3.2 Test II .....	259
3.3 Test III .....	259
4 Comparative Study .....	259
4.1 Comparative Test I .....	260
4.2 Comparative Test II .....	261
4.3 Comparative Test III .....	263
5 Conclusions .....	264
References .....	264
<b>hypDE: A Hyper-Heuristic Based on Differential Evolution for Solving Constrained Optimization Problems</b> .....	<b>267</b>
<i>José Carlos Villeda Tinoco, Carlos A. Coello Coello</i>	
1 Introduction .....	267
2 Differential Evolution .....	268
3 Hyper-Heuristics .....	270
4 Previous Related Work .....	270
5 Our Proposed Approach .....	271
5.1 Handling Constraints .....	275
6 Results .....	278
6.1 Analysis of Results .....	278
7 Conclusions and Future Work .....	280
References .....	281



## Part VII Evolutionary Computation for Vision, Graphics, and Robotics

### Evolutionary Computation Applied to the Automatic Design of Artificial Neural Networks and Associative Memories . . . . . 285

*Humberto Sossa, Beatriz A. Garro, Juan Villegas, Gustavo Olague, Carlos Avilés*

1	Introduction . . . . .	285
2	Basics on Artificial Neural Networks and Associative Memories . . . . .	286
	2.1 Basics on Artificial Neural Networks (ANNs) . . . . .	286
	2.2 Basics on Associative Memories (AMs) . . . . .	287
3	Automatic Synthesis of ANNs . . . . .	288
	3.1 PSO, DE and ABC . . . . .	289
	3.2 Garro's Proposal . . . . .	289
4	Automatic Synthesis of AMs by Means of Genetic Programming . . . . .	290
	4.1 Genetic Programming . . . . .	291
	4.2 Villegas' Proposal . . . . .	291
5	Experimental Results . . . . .	291
	5.1 Examples of Synthetically Generated ANNs . . . . .	292
	5.2 Examples of Synthetically Generated AMs . . . . .	292
6	Conclusions and Directions for Further Research . . . . .	295
	References . . . . .	295

### Segmentation of Blood Cell Images Using Evolutionary Methods . . . . . 299

*Valentín Osuna, Erik Cuevas, Humberto Sossa*

1	Introduction . . . . .	299
2	Gaussian Approximation and Otsu's Methods . . . . .	301
	2.1 Gaussian Approximation Method . . . . .	301
	2.2 Otsu's Method . . . . .	303
3	Differential Evolution and Artificial Bee Colony Optimization . . . . .	304
	3.1 Differential Evolution . . . . .	304
	3.2 Artificial Bee Colony Optimization . . . . .	305
4	Experimental Results . . . . .	306
5	Conclusions and Future Work . . . . .	309
	References . . . . .	309

### Fast Circle Detection Using Harmony Search Optimization . . . . . 313

*Erik Cuevas, Humberto Sossa, Valentín Osuna, Daniel Zaldivar, Marco Pérez-Cisneros*

1	Introduction . . . . .	313
2	Harmony Search Optimization . . . . .	315
3	Fitness Approximation Method . . . . .	317
4	Circle Detection Using HSO . . . . .	319
	4.1 Objective Function . . . . .	319

5	Final Detection Algorithm .....	320
6	Experimental Results .....	321
7	Conclusions .....	323
	References .....	324

**Particle Swarm Optimization Applied to Interferogram Demodulation ... 327**

*Julio Jiménez, Humberto Sossa, Francisco Cuevas*

1	Introduction .....	327
2	Particle Swarm Optimization .....	329
3	PSO Applied to Phase Recovery .....	330
4	PSOs Convergence .....	331
5	Experimental Results .....	331
6	Conclusions .....	335
	References .....	336

**Evolving Conspicuous Point Detectors for Camera Trajectory Estimation ... 339**

*Daniel Hernández, Gustavo Olague, Eddie Clemente, León Dozal*

1	Introduction .....	339
1.1	Visual Behavior .....	340
2	Evolutionary Visual Behavior .....	341
2.1	Conspicuous Point Detection .....	341
3	Trajectory Estimation with a SLAM System .....	344
4	Synthesis of Visual Behaviors .....	347
5	Experimental Results .....	349
6	Conclusions .....	351
7	Future work .....	352
	References .....	352

**Purposive Evolution for Object Recognition Using an Artificial Visual Cortex ... 355**

*Eddie Clemente, Gustavo Olague, Leon Dozal*

1	Introduction .....	355
2	Visual Cortex .....	357
3	Evolution and Teleology for Visual Processing .....	358
4	Artificial Visual Cortex (AVC) .....	359
4.1	Artificial Dorsal Stream (ADS) .....	361
4.2	Artificial Ventral Stream (AVS) .....	363
5	Evolving AVCs with Organic Genetic Programming .....	364
6	Experimental Results .....	365
7	Conclusions .....	368
	References .....	369

<b>Evolving an Artificial Dorsal Stream on Purpose for Visual Attention . . . .</b>	<b>371</b>
<i>León Dozal, Gustavo Olague, Eddie Clemente</i>	
1	Introduction . . . . . 371
1.1	Problem Statement . . . . . 372
2	Visual Attention Processing . . . . . 373
2.1	Classical Approach to Visual Attention . . . . . 373
2.2	An Unified Approach of Visual Attention . . . . . 375
3	Purposive Evolution for Visual Attention . . . . . 376
3.1	Acquisition of Early Visual Features . . . . . 376
3.2	Feature-Integration for Visual Attention . . . . . 378
4	Organic Genetic Programming . . . . . 379
5	Experiments and Results . . . . . 380
5.1	Evolution of VAPs for Aiming Scene Novelty . . . . . 380
5.2	Evolution of VAPs for Aiming Specific Targets . . . . . 382
6	Conclusions . . . . . 384
	References . . . . . 384
<b>Comparison of Two Evolutionary Approaches for the Polygonal Approximation of Digital Curves . . . . .</b>	<b>387</b>
<i>Paola Berenice Alvarado-Velazco, Victor Ayala-Ramirez</i>	
1	Introduction . . . . . 387
2	Polygonal Approximation of Digital Curves . . . . . 388
3	Polygonal Approximation of Digital Curves Using Evolutionary Meta-heuristics . . . . . 388
4	EP Approach to Approximate Digital Curves Using Straight Line Segments . . . . . 389
4.1	Individual Representation . . . . . 389
4.2	Mutation Operators . . . . . 389
4.3	Fitness Function . . . . . 390
5	Tests and Results . . . . . 391
5.1	Test Protocol . . . . . 391
5.2	Results . . . . . 392
6	Conclusions . . . . . 394
	References . . . . . 395
<b>Evolution of Contours for Topology Optimization . . . . .</b>	<b>397</b>
<i>Gideon Avigad, Erella Matalon Eisenstadt, Shaul Salomon, Frederico Gadelha Guimar</i>	
1	Background . . . . . 397
1.1	Evolutionary Algorithms . . . . . 398
1.2	EC for Size/Shape/Topology Optimization . . . . . 398
1.3	Niching within EC and for Topology Optimization . . . . . 399
1.4	Function Diversity . . . . . 400

2	Methodology .....	401
2.1	Describing Topologies by Functions .....	401
2.2	Problem Definition .....	403
2.3	Diverse Topologies and Niching .....	403
2.4	The Evolutionary Optimization Algorithm .....	405
3	Examples .....	407
3.1	Cross Section Optimization for a Structure Subjected to a Tensile Force .....	407
3.2	Cross Section Optimization for a Structure Subjected to a Moment .....	409
4	Summary and Conclusions .....	410
	References .....	410

**Part VIII Real-world Application of Bio-inspired Metaheuristics**

<b>Multi-Objective Particle Swarm Optimisation for Molecular Transition State Search .....</b>	<b>415</b>	
<i>Jan Hettenhausen, Andrew Lewis, Stephen Chen, Marcus Randall, René Fournier</i>		
1	Introduction .....	415
2	The Multi-Objective Optimisation Problem Formulation .....	419
3	Application of MOPSO .....	421
4	Results .....	424
5	Conclusion .....	427
	References .....	427

<b>A Multi-Objective Extremal Optimisation Approach Applied to RFID Antenna Design .....</b>	<b>431</b>	
<i>Pedro Gómez-Meneses, Marcus Randall, Andrew Lewis</i>		
1	Introduction .....	431
2	The Inseparable Fitness Evaluation Scheme .....	434
3	EO Applied to RFID Antenna Design .....	435
4	Computational Experiments .....	439
4.1	Comparison of Results .....	439
4.2	Manual Local Search Analysis .....	442
5	Conclusions .....	444
	References .....	445

<b>Wasp Colony with a Multiobjective Local Optimizer for Dynamic Task Planning in a Production Plant .....</b>	<b>447</b>	
<i>Luis Fernando Gutierrez-Marfileno, Eunice Ponce-de-Leon, Elva Diaz-Diaz, Leoncio Ibarra-Martinez</i>		
1	Introduction .....	447
2	Job Shop Scheduling Problem .....	449

3	Plant Description in Morley's Problem .....	450
4	Solution Proposal .....	451
4.1	Wasp Colony Algorithm .....	452
4.2	Routing Wasp Algorithm .....	452
4.3	Priority Rules for Scheduling .....	454
4.4	The Multiobjective Optimization Focus .....	454
4.5	System Modeling .....	455
5	Experimental Design .....	458
6	Conclusions .....	459
	References .....	460

### **Optimization Metaheuristic for Software Testing** ..... 463

*Nashat Mansour and Hratch Zeitunlian, Abbas Tarhini*

1	Introduction .....	463
2	Related Work on Testing Web Applications .....	464
3	Testing Web Applications .....	465
4	State Graph Modeling .....	466
4.1	State Graph .....	466
4.2	Building the State Graph .....	467
5	Simulated Annealing Algorithm .....	468
5.1	Solution Representation .....	469
5.2	Energy Function .....	469
5.3	Metropolis Step and Feasibility .....	470
5.4	Cooling Schedule .....	471
6	Experimental Results .....	471
7	Conclusions .....	473
	References .....	473

### **Stochastic Optimisation in Computational Engineering Design** ..... 475

*Timoleon Kipouros*

1	Introduction .....	475
2	Computational Design Approach .....	476
3	Indispensable Components of Automatic Design Systems .....	477
3.1	Parameterisation of the Design Problem .....	478
3.2	Evaluation and Modelling of the Objective Functions .....	479
3.3	Search Exploration of the Design Space .....	481
4	Contributions of Optimisation Intelligence .....	482
5	Auxiliary Tools .....	484
6	Grid Computing .....	486
7	Post-optimisation Analyses .....	486
8	Conclusions and Future Directions .....	488
	References .....	489

<b>Evolutionary Algorithms and Water Resources Optimization</b> .....	<b>491</b>
<i>Oluwatosin Olofintoye, Josiah Adeyemo, Fred Otieno</i>	
1 Introduction .....	491
2 Water Resources Management Using Genetic Algorithms .....	493
3 Water Resources Management Using Differential Evolution .....	496
4 Water Resources Management Using Evolution Strategy .....	498
5 Water Resources Management Using Genetic Programming .....	499
6 Conclusion .....	501
References .....	502
<b>Author Index</b> .....	<b>505</b>

**Part I**  
**Cell Mapping and Quasi-stationary**  
**Distributions**

# Control of Nonlinear Dynamic Systems with the Cell Mapping Method

Jian-Qiao Sun

**Abstract.** This paper studies control problems of nonlinear dynamic systems using the cell mapping method. We first present the formulation of optimal control problem and Bellman's principle of optimality. Then, we present the cell mapping methods and their application to optimal control problems of deterministic nonlinear dynamic systems. Examples of population dynamics control of two competing species are presented to demonstrate the effectiveness of the cell mapping method.

## 1 Introduction

Optimal control theory and its various applications have long been among the important topics of research in engineering. The optimal control problem can be solved by using Pontryagin's minimum principle and the Hamilton-Jacobi-Bellman equations (HJB). When the system is non-linear, and control and state constraints are imposed, finding solutions to optimal control problems becomes a very difficult task. The cell mapping method offers an attractive way to compute optimal control solutions in conjunction with Bellman's principle of optimality.

The cell mapping methods were first introduced by Hsu [16, 18] to study the global dynamics of nonlinear systems. Two cell mapping methods have been extensively studied, namely, the simple cell mapping and the generalized cell mapping. The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems [17, 3, 11]. Other interesting applications of the cell mapping methods include optimal space craft momentum unloading [13], single and multiple manipulators of robots [29], optimum trajectory planning in robotic systems [25], and tracking control of the read-write head of computer hard

---

Jian-Qiao Sun

School of Engineering, University of California, Merced, CA 95343, U.S.A.

e-mail: [jqsun@ucmerced.edu](mailto:jqsun@ucmerced.edu)



disks [26]. Sun and his group studied the fixed final state optimal control problems with the simple cell mapping method [7, 6], and applied the cell mapping methods to the optimal control of deterministic systems described by Bellman's principle of optimality [10]. Crespo and Sun further applied the generalized cell mapping based on the short-time Gaussian approximation to stochastic optimal control problems [9, 11]. They also studied semi-active optimal control of populations of competing species in a closed environment with the cell mapping method [8].

We study the deterministic optimal control in this paper. In Section 2, we review the formulation of optimal control problems and Bellman's principle of optimality. Section 3 presents the cell mapping methods and their application to the optimal control problem. Section 4 studies optimal control problems of competing species in a closed habitat. Section 5 concludes the paper.

## 2 Optimal Control

Consider a nonlinear dynamic system,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t, \mathbf{u}), \quad (1)$$

where  $\mathbf{x}(t) \in \mathbf{R}^n$  is the state vector,  $\mathbf{u}(t) \in \mathbf{R}^m$  the control input and  $\mathbf{x}(t_0) = \mathbf{x}_0$  is the initial condition. Define the performance index,  $J$  as:

$$J(\mathbf{u}, \mathbf{x}_0, t_0, T) = \phi(\mathbf{x}_T, T) + \int_{t_0}^T L(\mathbf{x}(t), \mathbf{u}(t)) dt, \quad (2)$$

where  $t \in [t_0, T]$  is the time interval of interest,  $\phi(\mathbf{x}_T, T)$  is the terminal cost at the state  $\mathbf{x}_T = \mathbf{x}(T)$ , and  $L(\mathbf{x}(t), \mathbf{u}(t))$  is the Lagrangian function. Here, we assume that the Lagrangian function is not an explicit function of time. The optimal control problem is to find a control  $\mathbf{u}(t)$  within a set  $\mathbf{U} \subset \mathbf{R}^m$  on the time interval  $[t_0, T]$  that drives the system from the initial condition to the target set defined by  $\Psi(\mathbf{x}_T, T) = 0$  such that the cost function  $J$  is minimized. If the Lagrangian function satisfies convexity conditions, the optimal control solution can be found via Pontryagin's minimum principle or the Hamilton-Jacobi-Bellman (HJB) equations.

Bellman's principle of optimality was originally stated as follows [1, 2]: “An optimal policy has the property that no matter what the previous decisions have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from these previous decisions”. We restate Bellman's principle as follows: Let  $(\mathbf{x}^*, \mathbf{u}^*)$  be an optimal control solution pair over the time interval  $[t_0, T]$  subject to the initial condition  $\mathbf{x}(t_0) = \mathbf{x}_0$ . Let  $\hat{t}$  be a time instant such that  $t_0 \leq \hat{t} \leq T$ . Then,  $(\mathbf{x}^*, \mathbf{u}^*)$  is still the optimal control solution pair from  $[\hat{t}, T]$  subject to the initial condition  $\mathbf{x}(\hat{t}) = \mathbf{x}^*(\hat{t})$ .

We shall combine Bellman's principle of optimality with the cell mapping method to obtain optimal controls. When the number of control actions is finite, Bellman's principle provides a basis for the search of optimal controls. It also implies that the search for optimal controls should be done backward in time starting from the target state.

Let  $V(\mathbf{x}_0, t_0, T) = J(\mathbf{u}^*, \mathbf{x}_0, t_0, T)$  be the so-called value function or optimal cost function. Bellman's principle of optimality can be stated as [27]

$$\begin{aligned} V(\mathbf{x}_0, t_0, T) &= \inf_{\mathbf{u} \in \mathbf{U}} \left( \int_{t_0}^{\hat{t}} L(\mathbf{x}(t), \mathbf{u}(t)) dt + \int_{\hat{t}}^T L(\mathbf{x}(t), \mathbf{u}(t)) dt + \phi(\mathbf{x}_T, T) \right) \\ &= \inf_{\mathbf{u} \in \mathbf{U}} \left( \int_{t_0}^{\hat{t}} L(\mathbf{x}(t), \mathbf{u}(t)) dt + V(\mathbf{x}_{\hat{t}}, \hat{t}, T) \right), \end{aligned} \quad (3)$$

where  $t_0 \leq \hat{t} \leq T$ ,  $\mathbf{x}_{\hat{t}} = \mathbf{x}(\hat{t})$  and  $V(\mathbf{x}_T, T, T) = \phi(\mathbf{x}_T, T)$ .

Consider the optimal control of a system starting from  $\mathbf{x}_i$  in the time interval  $[i\tau, T]$  where  $\tau$  is a discrete time step. Define an incremental cost and an accumulative cost as

$$J_\tau = \int_{i\tau}^{(i+1)\tau} L(\mathbf{x}(t), \mathbf{u}(t)) dt, \quad (4)$$

$$J_T = \phi(\mathbf{x}_T^*, T) + \int_{(i+1)\tau}^T L(\mathbf{x}^*(t), \mathbf{u}^*(t)) dt. \quad (5)$$

In  $J_T$ ,  $(\mathbf{x}^*(t), \mathbf{u}^*(t))$  is the optimal solution pair over the time interval  $[(i+1)\tau, T]$ . Then, Bellman's principle of optimality can be restated as:

$$\begin{aligned} V(\mathbf{x}_i, i\tau, T) &= \inf_{\mathbf{u} \in \mathbf{U}} \{J_\tau + J_T\} \\ &= \inf_{\mathbf{u} \in \mathbf{U}} \{J_\tau + V(\mathbf{x}_{i+1}, (i+1)\tau, T)\}. \end{aligned} \quad (6)$$

The incremental cost is the cost for the system to march one step forward starting from the initial condition  $\mathbf{x}_i = \mathbf{x}(i\tau)$ . The system lands on the intermediate set of the state variables  $\mathbf{x}_{i+1} = \mathbf{x}((i+1)\tau)$ . The accumulative cost is the cost for the system to reach the target set starting from this intermediate set, and is calculated through the accumulation of incremental costs over several short time intervals between  $(i+1)\tau$  and  $T$ .

Bellman's principle of optimality as stated in Equation (6) suggests that one can obtain a local solution of the optimal control problem over a short time interval  $\tau$  to form the global solution provided that  $\mathbf{x}_{i+1}$  lies on the optimal solution. The global solution consists of all the local solutions that are constructed backward in time starting from the terminal condition  $\phi(\mathbf{x}_T, T)$  at time  $T$ . The cell mapping method offers an effective way to compute all the local optimal solutions from Equation (6) in a given region.

### 3 Cell Mapping Methods

The cell mapping methods transform the point-to-point dynamics into a cell-to-cell mapping by discretizing both phase space and the integration time. We denote all admissible mapping time steps as  $\mathbb{T} = \{\Delta t_1, \Delta t_2, \dots, \Delta t_N\}$ . The point-to-point mapping obtained from Equation (1) is given by

$$\mathbf{x}(k) = \mathbf{F}(\mathbf{x}(k-1), \mathbf{u}(k), \Delta t(k)), \quad (7)$$

where  $\mathbf{x}(k) \in \mathbf{R}^n$  is the state vector at the  $k^{\text{th}}$  mapping step and  $\mathbf{u}(k) \in \mathbb{U}$  is the control in the  $k^{\text{th}}$  time interval. By making  $\mathbf{u}(k) = 0$  in Equation (7), the uncontrolled system dynamics can be studied. In the simple cell mapping (SCM), the dynamics of an entire cell denoted as  $Z$  is represented by the dynamics of its center. The center of  $Z$  is mapped according to the point-to-point mapping. The cell that contains the image point is called the image cell of  $Z$ . The cell-to-cell mapping is denoted by  $C$ ,

$$Z(k) = C(Z(k-1), \mathbf{u}(k), \Delta t(k)). \quad (8)$$

Consequently, the exact image of the center of  $Z$  is approximated by the center of its image cell. This approximation can cause significant errors in the long term solution of the control problem computed by the SCM method [19, 17, 3, 25].

Since the image of the entire cell covers a bounded region, more than one cell can be the image cells. If we don't restrict the number of image cells of a pre-image cell to be one, we come to the generalized cell mapping (GCM) [18]. Under GCM, a cell can have several images, everyone of which has certain probability to be part of the system solution. The GCM method provides a probabilistic description of the system response. The evolution of the system dynamics is governed by a finite Markov chain. The transition probability matrix of the Markov chain contains the topological structure of the global system response in the cell state space. Attractors, basins of attraction and separatrices may be identified by finding the limiting probability distribution and the corresponding transient and persistent groups of cells. For more discussions on the cell mapping methods, the reader is referred to [18].

#### 3.1 Control Application

When applying the cell mapping method to the optimal control problem, we need to construct a database of cell mappings under all allowable controls. We denote the set of discrete admissible controls as  $\mathbb{U}$ . Let  $N_c$  be the number of admissible controls in the set  $\mathbb{U}$ . Let  $N_m$  denote the set of mappings from a pre-image cell to its first  $N_m$  consecutive image cells along the trajectory under a given control. These mappings can have non-uniform mapping time steps so that the images of the mapping are closest to the center of the cell containing the image [7].

The control database consists of the following elements: for each pre-image cell  $\mathbf{z}_i$ , there are  $N_c \times N_m$  image cells  $\mathbf{z}_j$ , the corresponding controls  $\mathbf{u}_l$ , the associated mapping time steps  $\Delta t_{ijl}$  and the incremental control costs

$\Delta J_{ijl} = \int_{t_0}^{t_0 + \Delta t_{ijl}} L(\mathbf{x}(t), \mathbf{u}(t), t) dt$ .  $\mathbf{z}_i$  denotes the integer coordinates of the  $i^{\text{th}}$  cell. We denote the complete set of mappings by  $\mathbf{M}$ . A special subset of  $\mathbf{M}$  denoted by  $\mathbf{N}$  contains the image cells in the closest neighborhood of every pre-image cell  $\mathbf{z}_i$  under all admissible controls. This group of neighborhood image cells consists of the set of adjacent cells surrounding the pre-image cell only.

Let  $\Omega \in \mathbf{R}^n$  denote the set of cells representing the target set defined by  $\Psi(\mathbf{x}_T, T) = 0$ . The iterative backward search algorithm is described here. Let  $N_b$  be the number of backward search iterations that we would like to carry out. Initially, the search is over the mapping set  $\mathbf{N}$  according to the following steps.

- (i) Identify the cells that are mapped into  $\Omega$  in one step.
- (ii) Assign a cumulative cost to each cell found in Step [\(i\)](#). The cumulative cost is the smallest cost for the system to move from the current cell to the original target set. It is calculated by adding the cumulative cost of the image cell and the incremental cost of the current cell. If more than one image cells are involved, the smallest cumulative cost is taken. Note that the cells in the original target set have zero cumulative costs.
- (iii) Expand the target set  $\Omega$  by including all the cells found in Step [\(i\)](#) with the minimal cost.
- (iv) Repeat the search from Step [\(i\)](#) until all the cells in the state space are processed.
- (v) Examine the cumulative costs of all  $N_m$  consecutive image cells  $\mathbf{z}_j$  for every pre-image cell  $\mathbf{z}_i$  and for every control  $\mathbf{u}_l$  in  $\mathcal{U}$ . We retain the image cell that has the smallest cumulative cost. This cell is stored in a set  $\mathbf{M}^*$ .
- (vi) Repeat from Step [\(i\)](#) over the set  $\mathbf{M}^*$ .

After going through the above backward search  $N_b$  times, degenerated control solutions are generally eliminated. It has been shown in [\[7\]](#) that the average cost over the entire phase space is reduced with the number of backward searches. In other words, the above backward search is a converging algorithm.

Note that the final set  $\mathbf{M}^*$  contains the information on the location of the switching curves and the optimal controls for each cell. A discriminating function which considers smoothness and local continuity of the trajectories is used to break the cost ties in the backward searches.

## 4 Optimal Control of Competing Species

In this paper, we consider examples of optimal control of competing species. There are many studies of mathematical models of population dynamics in the literature [\[21, 23, 12, 22\]](#). The classical Lotka-Volterra models for competing species [\[20, 28\]](#) and for the predator-prey interaction [\[15, 14\]](#) are two well known examples. In some

optimal control problems, the inclusion of dispersal processes and time delays in autonomous and non-autonomous models are used as control means [28, 15]. The use of performance indices that promote simultaneously economic and ecologic goals are studied in [24]. The optimal birth control of age-dependent models is considered in [5, 4]. The parametric control of the interaction factors between species is investigated in [15]. It should be pointed out that many optimal control problems of population dynamics involve parametric manipulations that are highly nonlinear. We apply the simple cell mapping method to solve the optimal control problem of population dynamics of two competing species described by the Lotka-Volterra model.

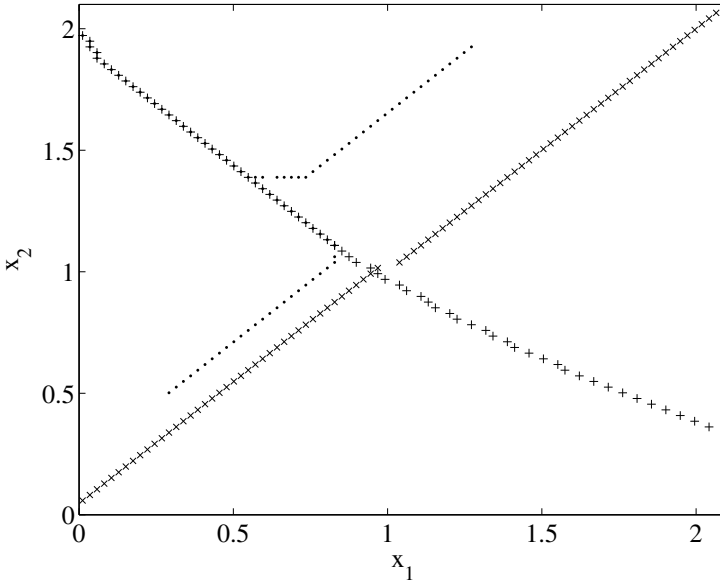
The classical Lotka-Volterra model describes the interaction between two species competing for an essential and limited resource in a closed habitat. All other external influences such as predators and seasonal effects are neglected. According to the Lotka-Volterra model, each species would grow to its carrying capacity in the absence of the other. The population dynamics is described by the logistic growth for each species. The interaction between the species is modeled as the conflict that occurs at a rate proportional to the size of each population. The Lotka-Volterra model is described by the following set of coupled nonlinear ordinary differential equations,

$$\begin{aligned}\dot{x}_1 &= x_1(p_1 - s_1x_1 - c_1x_2), \\ \dot{x}_2 &= x_2(p_2 - s_2x_2 - c_2x_1),\end{aligned}\tag{9}$$

where  $x_1$  and  $x_2$  are the population densities of two species and the coefficients  $p_i$ ,  $s_i$  and  $c_i$  ( $i = 1, 2$ ) are the intrinsic growing rates, the saturation factors and the interaction coefficients, respectively.

The system described by Equation (9) has two stable fixed points at  $[0, p_2/s_2]$  and  $[p_1/s_1, 0]$ , an unstable node at  $[0, 0]$ , and a saddle node at  $[(p_1 - c_1((p_1c_2 - p_2s_1)/(c_1c_2 - s_1s_2)))/s_1, (p_1 - c_1((p_1c_2 - p_2s_1)/(c_1c_2 - s_1s_2)))/s_1]$ . In the phase space, the unstable manifold of the saddle connects the stable points and the stable manifold defines the boundary of the basin of attraction of the two stable states (see Figure 1). It is important to mention that when the system is under parametric controls, the basin of attraction changes its topology. The boundaries of the basin partition the phase space into regions of initial conditions that lead to different long-term solutions. In the present study, they define the controllable region for a given control strategy. The search algorithm for optimal controls in the cell state space can also delineate the manifolds of the saddle node and the domains of attraction/repulsion of the fixed points of the nominal system.

Recall that the manifolds of the saddle point define the boundaries of the basins of attraction of the stable node  $[0, p_2/s_2]$  in the forward mapping, and of the unstable node  $[0, 0]$  in the backward mapping. Specifically, we apply the algorithm to find these manifolds in the phase space by treating the stable node  $[0, p_2/s_2]$  and the unstable node  $[0, 0]$  as the final state of a fictitious optimal control problem.



**Fig. 1** Location of the stable ( $\times$ ) and unstable ( $+$ ) manifolds of the unstable saddle found by the cell mapping method. The evolution of populations starting from two initial conditions without control is marked as dots.

#### 4.1 Numerical Examples of Optimal Control

Let  $p_1 = 3$ ,  $p_2 = 2$ ,  $s_1 = 1$ ,  $s_2 = 1$ ,  $c_1 = 2$ , and  $c_2 = 1$ . We cover the region  $[0, 2.1] \times [0, 2.1]$  in the state space with 8100 square cells. A uniform time step is used in this work. The unstable ( $+$ ) and stable ( $\times$ ) manifolds of the saddle node found by the cell mapping are shown in Figure 1.

Figure 1 also shows the population evolution starting from two different initial conditions in the basin of attraction of the stable node  $[0, p_2/s_2]$ . The trajectories are marked with dots. All initial conditions located above the stable manifold of the saddle will move to  $[0, p_2/s_2]$  in the long term, and the initial conditions below this manifold will move to  $[p_1/s_1, 0]$ . The dichotomy implied by the basin boundary occurs in other population models of competition and has led biologist to formulate the principle of competitive exclusion, which states that two species competing for the same limited resource cannot coexist. Analysis and discussions about the biological interpretation of this phenomena can be found in [23, 12, 22].

Different control problems are studied numerically in this paper. In the example, we have chosen  $\phi(\mathbf{x}_T, T) = 0$  and  $L(\mathbf{x}(t), \mathbf{u}(t), t) = 1$ , which defines a minimum time problem. In the first example, the fixed final state is chosen to be the fixed point  $[0, p_2/s_2]$  where species  $x_1$  is extinct in the long term. In the second example, we want to drive the system to the location of the nominal saddle node

$[(p_1 - c_1((p_1 c_2 - p_2 s_1)/(c_1 c_2 - s_1 s_2)))/s_1, (p_1 - c_1((p_1 c_2 - p_2 s_1)/(c_1 c_2 - s_1 s_2)))/s_1]$  where both species coexist, that is, we stabilize the unstable saddle point.

### Extinction of One Species

We consider to extinct one species in minimum time by adding the population of the other from an external habitat. This approach is common to animal population control in the wild. Assume that a fixed number of species  $x_2$  can be brought into the habitat at a time. Let  $u > 0$  be the population density to be added to  $x_2$  artificially in the habitat. The state equation (9) is modified to be

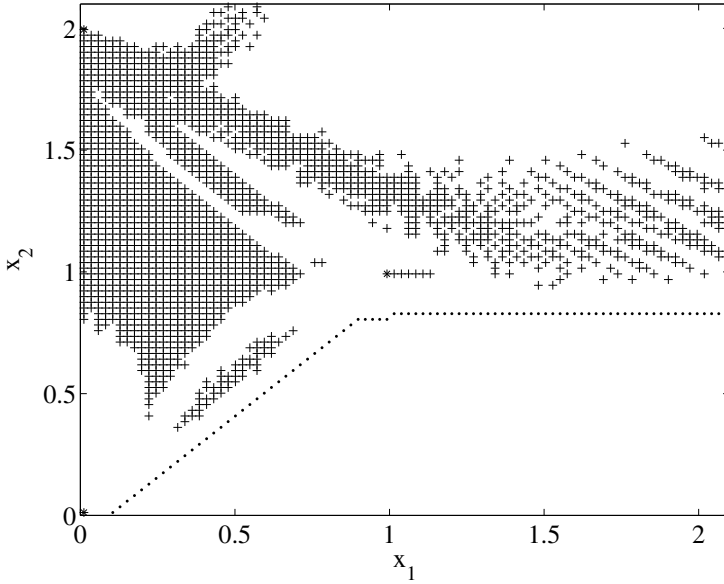
$$\begin{aligned}\dot{x}_1 &= x_1(p_1 - s_1 x_1 - c_1(x_2 + u)), \\ \dot{x}_2 &= (x_2 + u)(p_2 - s_2(x_2 + u) - c_2 x_1).\end{aligned}\tag{10}$$

In practice, the inclusion of new population of a species to the habitat is often performed at discrete time instants only. In the cell mapping,  $u$  is implemented at the beginning of each mapping step. Mathematically, this control can be expressed as a sequence of impulsive inputs at the time instants of mappings. Such a control is equivalent to a change of the initial condition of the system at the beginning of the mapping step.

We choose a bi-level control set as  $\mathbb{U} = \{0, 0.2\}$ , i.e.  $u = 0$  or  $u = 0.2$ . Figure 2 shows the optimal control solution for the final state located at  $\mathbf{x}_T = (0, 2)$ . The cells marked by the dots represent the boundary of the controllable region. Below this boundary, the system cannot reach the prescribed target with the given bi-level control set. The cells marked with  $\times$  above the boundary represent the areas where the optimal control is passive with  $u = 0$ . The unmarked cells above the basin boundary represent the cells where the optimal control is active with  $u = 0.2$ . All the cells above the basin boundary form the controllable region, meaning that an optimal control sequence in  $\mathbb{U}$  can be found to lead any initial condition in the region to the final state in minimum time. The boundary of the areas with the same optimal control in the controllable region appears to be fairly complex as shown in Figure 2.

The controllable region shown in Figure 2 is bigger than the domain of attraction of the stable node at  $(0, 2)$  of the uncontrolled system shown in Figure 1. The control with  $u > 0$  displaces the saddle point *downward* along a vertical line in the phase space. The basin of attraction of the final state, i.e. the stable node  $(0, 2)$ , is thus enlarged.

Figure 3 shows the optimal control solution as the function of time for an initial condition at  $\mathbf{x}_0 = (0.5, 0.6)$ . The upper figure shows the evolution of populations. The lower figure presents the sequence of optimal control actions. The optimal control is generally a combination of active and passive actions. Figure 4 shows the optimal trajectory of the populations in the phase space. In this case, the nominal system without control starting from this initial condition reaches the target cell in 13.17 time units. Under the optimal control, the time is reduced to 8.56 units.



**Fig. 2** Controllable region and optimal controls in the phase space for extinguishing the species  $x_1$  by including animals of the species  $x_2$  from an external habitat. The cells above the boundary marked by dark dots form the controllable region. The cells marked with + represent the area with passive control  $u = 0$ , and the unmarked cells above the boundary represent the area with active control  $u = 0.2$ .

Figure 5 shows the optimal trajectories starting from four initial conditions in the phase space. Two of these four initial conditions,  $(0.3, 0.5)$  and  $(1.3, 1.95)$ , have been studied in Figure 1 without control. By comparing the trajectories starting from these two initial conditions, one can see that the optimal control moves the state to the target cell faster. In fact, the time for these two initial conditions to reach the target without control are 10.98 and 7.26 units, and 6.52 and 6.55 units with control. The other two initial conditions cannot reach the target without control.

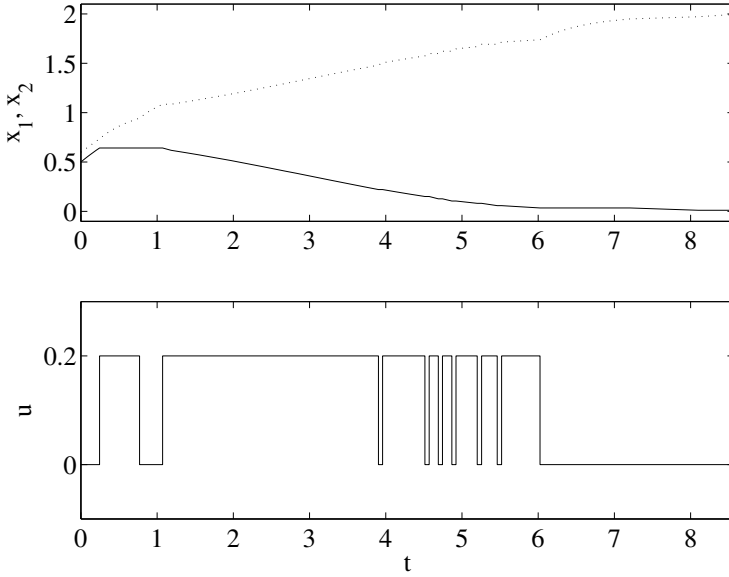
Finally, we note that Equation (10) is nonlinear and is not in the control affine form either. The analytical solutions of the control problem for such a system are thus quite difficult to obtain.

Another way to control the population is to change the intrinsic growing rate of the undesirable species. An example is to control the intrinsic growing rate  $p_1$  of  $x_1$  as follows,

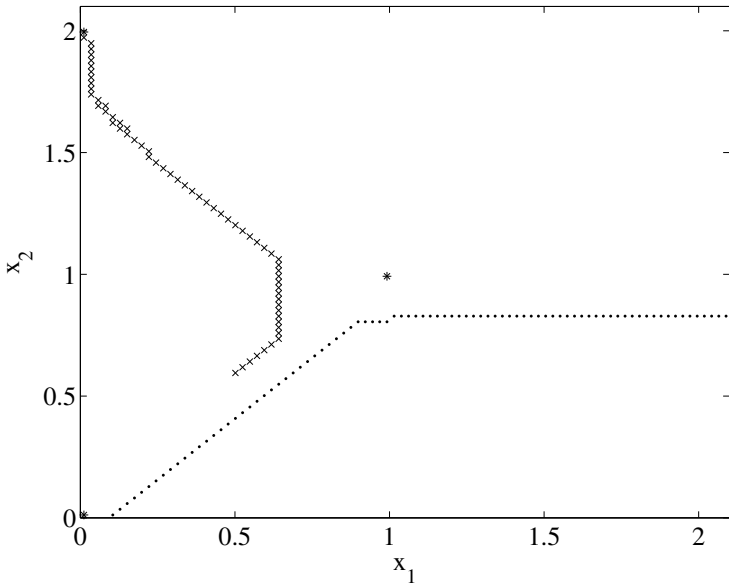
$$\begin{aligned}\dot{x}_1 &= x_1(p_1(1-u) - s_1x_1 - c_1x_2), \\ \dot{x}_2 &= x_2(p_2 - s_2x_2 - c_2x_1).\end{aligned}\quad (11)$$

More numerical results of the optimal control can be found in [8].

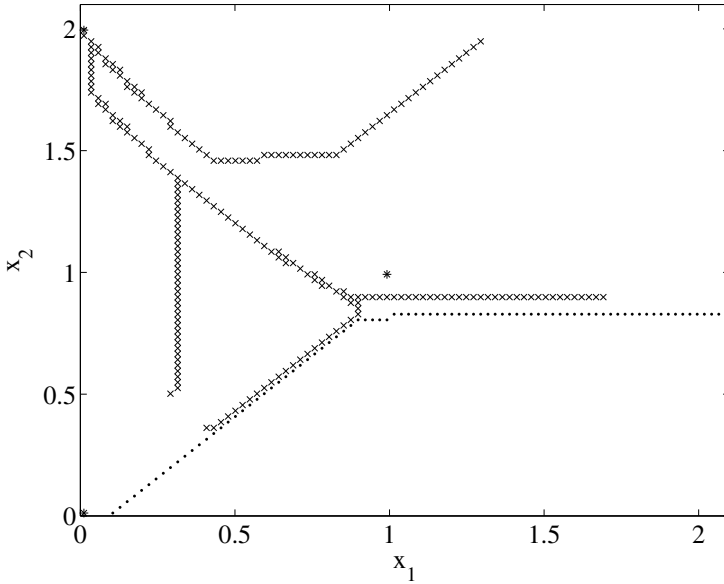




**Fig. 3** Population evolutions (top) and optimal control (bottom) in the time domain for the problem studied in the first example. In the top figure, solid line represents  $x_1$  and dashed line represents  $x_2$ .



**Fig. 4** The optimal population trajectory from  $\mathbf{x}_0 = (0.5, 0.6)$  in the phase space. The boundary of the controllable region is marked by dark dots.



**Fig. 5** The optimal population trajectories ( $\times$ ) for four different initial conditions in the phase space. Two initial conditions are studied in Figure 4. The boundary of the controllable region of the target cell is marked by dark dots.

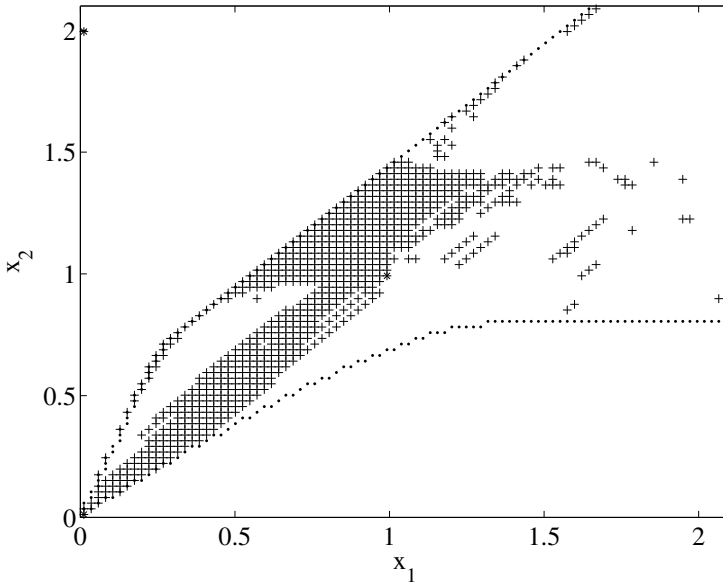
### Coexistence of Two Species

Let us now consider a control to maintain the coexistence of both species by adjusting the coupling parameter of the system. The parametric control modifies the state equations to be

$$\begin{aligned}\dot{x}_1 &= x_1(p_1 - s_1x_1 - (c_1 + u)x_2), \\ \dot{x}_2 &= x_2(p_2 - s_2x_2 - (c_2 - u)x_1).\end{aligned}\quad (12)$$

The admissible control set is bi-level with  $\mathbb{U} = \{-0.1, 0.1\}$ . The final state of the system is chosen to be  $\mathbf{x}_T = (1, 1)$  at the saddle node of the nominal system. Figure 6 shows the controllable region with the given bi-level control set and the optimal controls in the phase space. The controllable region has a lower and upper boundary. These boundaries are marked by the dark dots. The cells marked with  $\times$  in the controllable region represent the areas where the optimal control takes the value  $u = -0.1$  and the unmarked cells in the controllable region represent the areas where the control is  $u = 0.1$ . The area below and above the boundaries represents the uncontrollable region.

Figure 7 shows the optimal control solution starting from an initial condition at  $\mathbf{x}_0 = (0.3, 0.5)$ . The upper figure shows the evolution of populations. The lower



**Fig. 6** Controllable region and optimal controls in the phase space for coexistence of species  $x_1$  and  $x_2$  by varying the coupling parameter between the two species. Initial conditions located outside the region between the boundaries marked by dark dots cannot reach the final state. Areas marked with + represent cells with control  $u = -0.1$ , and the unmarked cells between the boundaries represent the area with  $u = 0.1$ .

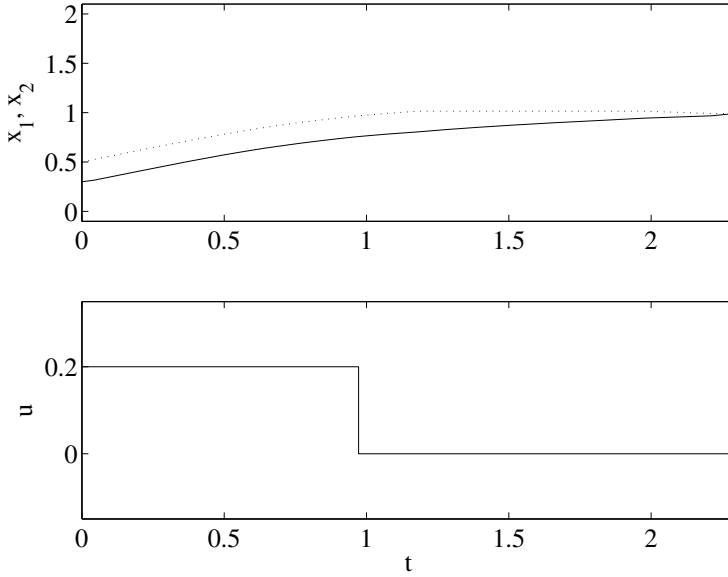
figure presents the sequence of optimal controls needed to drive the system to the final state. Figure 8 shows the optimal trajectory of populations in the phase space.

### A Remark

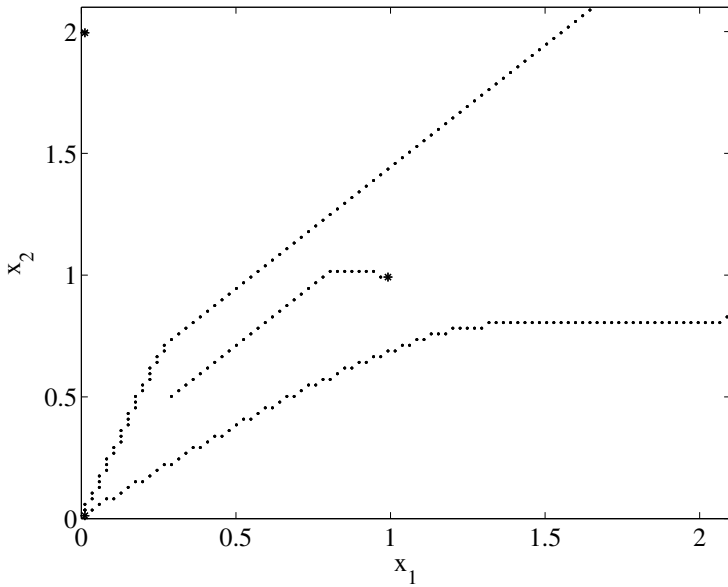
Can the cell mapping based control strategy achieve different objectives? For example, can we achieve the coexistence of two species by adding population of one species from an external habitat or by changing the intrinsic growing rates? Can we wipe out one population by varying the coupling parameters? It turns out that the answers are positive.

Figure 9 shows an example of the controllable region for achieving the coexistence of the two species at  $(1.1, 0.9)$  by using the same control strategy in Equation (11). We found that, similar to the results shown in Figure 6, there is a region in the phase space where the control can keep both species alive.

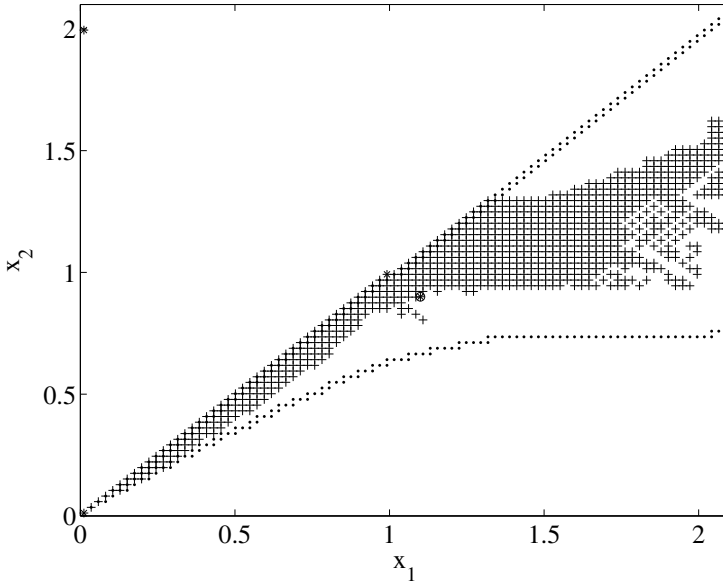
In all the cases studied, we can view the control as a means to move the fixed points of the system and change their corresponding basins of attraction in order to achieve control objectives. Such a phenomenon can be studied easily by the cell mapping method. In this paper, we have selected the admissible control set  $\mathbb{U}$  in such



**Fig. 7** Population evolutions (top) and optimal control (bottom) in the time domain for the problem studied in the second example. In the top figure, solid line represents  $x_1$  and dashed line represents  $x_2$ .



**Fig. 8** The optimal population trajectory in the phase space for the problem studied in the second example starting from  $\mathbf{x}_0 = (0.3, 0.5)$ .



**Fig. 9** Controllable region and optimal controls in the phase space for coexistence of both species by varying the intrinsic growing rate of  $x_1$ . Initial conditions located below and above the boundary marked by dark dots cannot reach the final state at  $(1.1, 0.9)$  in the center of the figure. Areas marked with + represent cells with passive control  $u = 0$ , and the unmarked cells above the boundary represent the area with active control  $u = 0.1$ .

a way that the system does not undergo bifurcations for all controls  $u \in \mathbb{U}$  while the fixed points preserve their nominal stability. If  $u$  is allowed to induce bifurcations, the geometry of the domains of attractions will change drastically when the control changes slightly.

## 5 Conclusions

We have presented studies of optimal control problems of a nonlinear system governing the population dynamics of two species with the cell mapping method. The optimal control is computed with the help of Bellman's principle of optimality. Numerical examples of the optimal control of population dynamics of two competing species are presented to demonstrate the effectiveness of the proposed method. We should point out that the proposed method has potential to be able to design controls for fairly complex nonlinear dynamic systems.

## References

1. Bellman, R.: On the theory of dynamic programming. *Proceedings of the National Academy of Science* 38, 716–719 (1952)
2. Bellman, R.: *Dynamic Programming*. Princeton University Press, Princeton (1957)
3. Bursal, F.H., Hsu, C.S.: Application of a cell-mapping method to optimal control problems. *International Journal of Control* 49(5), 1505–1522 (1989)
4. Chan, W.L., Zhu, G.B.: Optimal birth control of population dynamics II. problems with free final time, phase constraints, and min-max costs. *Journal of Mathematical Analysis and Applications* 146, 523–539 (1990)
5. Chan, W.L., Zhu, G.B.: Overtaking optimal control problem of age-dependent populations with infinite horizon. *Journal of Mathematical Analysis and Applications* 150, 41–53 (1990)
6. Crespo, L.G., Sun, J.Q.: Optimal control of target tracking via simple cell mapping. *Journal of Guidance and Control* 24, 1029–1031 (2000)
7. Crespo, L.G., Sun, J.Q.: Solution of fixed final state optimal control problems via simple cell mapping. *Nonlinear Dynamics* 23, 391–403 (2000)
8. Crespo, L.G., Sun, J.Q.: Optimal control of populations of competing species. *Nonlinear Dynamics* 27, 197–210 (2002)
9. Crespo, L.G., Sun, J.Q.: Stochastic optimal control of non-linear systems via short-time Gaussian approximation and cell mapping. *Nonlinear Dynamics* 28, 323–342 (2002)
10. Crespo, L.G., Sun, J.Q.: Fixed final time optimal control via simple cell mapping. *Nonlinear Dynamics* 31, 119–131 (2003)
11. Crespo, L.G., Sun, J.Q.: Stochastic optimal control of nonlinear dynamic systems via bellman's principle and cell mapping. *Automatica* 39(12), 2109–2114 (2003)
12. Edelstein-Keshet, L.: *Mathematical Models in Biology*. Random House, New York (1988)
13. Flashner, H., Burns, T.F.: Spacecraft momentum unloading: the cell mapping approach. *Journal of Guidance, Control and Dynamics* 13, 89–98 (1990)
14. Hironori, H.: Optimal control of nonlinear population dynamics. *IEEE Transactions on Systems, Man and Cybernetics* SMC-10(1), 32–38 (1980)
15. Hrinca, I.: An optimal control problem for the Lotka-Volterra system with delay. *Nonlinear Analysis, Theory, Methods and Applications* 28(2), 247–262 (1997)
16. Hsu, C.S.: A theory of cell-to-cell mapping dynamical systems. *Journal of Applied Mechanics* 47, 931–939 (1980)
17. Hsu, C.S.: A discrete method of optimal control based upon the cell state space concept. *Journal of Optimization Theory and Applications* 46(4), 547–569 (1985)
18. Hsu, C.S.: *Cell-to-cell Mapping, A Method of Global Analysis for Nonlinear Systems*. Springer, New York (1987)
19. Hsu, C.S., Chiu, H.M.: Global analysis of a system with multiple responses including a strange attractor. *Journal of Sound and Vibration* 114(2), 203–218 (1987)
20. Kiyotaka, I., Hashem, M.M.A., Keigo, W.: Evolution strategy with competing subpopulations. In: *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, New York, pp. 306–311 (1997)
21. Kolosov, G.E.: On a problem of control of the size of populations. *Journal of Computer and Systems Sciences International* 34(4), 115–122 (1996)
22. Murray, J.: *Mathematical Biology*. Springer, New York (1989)
23. Pianka, E.R.: *Competition and Niche Theory*. Sinauers Associates, Sunderland (1981)

24. Suzanne, L.: Optimal control of boundary habitat hostility for interacting species. *Mathematical Methods in Applied Sciences* 22(13), 1061–1077 (1999)
25. Wang, F.Y., Lever, P.J.A.: A cell mapping method for general optimum trajectory planning of multiple robotic arms. *Robotics and Autonomous Systems* 12, 15–27 (1994)
26. Yen, J.Y.: Computer disk file track accessing controller design based upon cell to cell mapping. In: *Proceedings of The American Control Conference* (1992)
27. Yong, J., Zhou, X.Y.: *Stochastic Controls, Hamiltonian Systems and HJB Equations*. Springer, New York (1999)
28. Zhang, J., Chen, L., Che, X.D.: Persistence and global stability for two species nonautonomous competition Lotka-Volterra patch-system with time delay. *Nonlinear Analysis* 37, 1019–1028 (1999)
29. Zhu, W.H., Leu, M.C.: Planning optimal robot trajectories by cell mapping. In: *Proceedings of Conference on Robotics and Automation*, pp. 1730–1735 (1990)

# Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions

Jose Blanchet, Peter Glynn, and Shuheng Zheng

**Abstract.** This paper studies a method for estimating the quasi-stationary distribution of various interacting particle processes has been proposed by [6, 5, 8]. This method improved upon existing methods in eigenvector estimation by eliminating the need for explicit transition matrix representation and multiplication. However, this method has no firm theoretical foundation. Our paper analyzes the algorithm by casting it as a stochastic approximation algorithm (Robbins-Monro) [12]. In doing so, we prove its convergence and rate of convergence. Based on this insight, we also give an example where the rate of convergence is very slow. This problem can be alleviated by using an improved version of this algorithm that is given in this paper. Numerical experiments are described that demonstrate the effectiveness of this improved method.

**Keywords:** Stochastic Approximation, Quasi-stationary Distributions.

## 1 Introduction

The original motivation for this algorithm came from physicists' need to estimate the quasi-stationary distribution of the contact process [6, 5, 8]. A quasi-stationary distribution can be computed via the left principal eigenvector of the transition matrix (transition rate matrix in the continuous-time setting). The method that has been proposed by these physicists is a heuristic based on manipulation of the Kolmogorov

---

Jose Blanchet · Shuheng Zheng  
Columbia University

e-mail: [jose.blanchet@ieor.columbia.edu](mailto:jose.blanchet@ieor.columbia.edu),  
[JohnZ622@gmail.com](mailto:JohnZ622@gmail.com)

Peter Glynn  
Stanford University

e-mail: [glynn@stanford.edu](mailto:glynn@stanford.edu)



forward equations. The method works in practice but has no firm proof. This paper recognizes the algorithm as a stochastic approximation algorithm which allows us to prove convergence and sufficient conditions for a Central Limit Theorem. We then give an improved version with variance reduction.

This section reviews the relevant related literature on eigenvector estimations. Sect. 2 reviews some background material to the contact process, quasi-stationary distributions, and the basis for the original heuristic. Sect. 3 goes over the stochastic approximation formulation and sketches the proof of convergence (the full proof will be given in a follow-up journal paper [1]). Sect. 4 gives an improved version of the algorithm. Sect. 5 studies the algorithm adapted for continuous-time Markov chains. Sect. 6 goes over several important numerical experiments.

## 1.1 Related Literature

### Power Method

The power method [9] is very simple. We iterate a sequence  $x_n$  by computing

$$x_{n+1} = \frac{x_n^T A}{\|x_n^T A\|}$$

This works for any matrix such that the principal eigenvalue has multiplicity one and strictly largest magnitude. The problem is that for Markov chains with extremely large state space, such as the contact process, it would not be feasible to store and compute in such large dimensions (on the order of  $2^n$  for interacting particle systems).

The variant known as inverse method also suffers from similar problems due to the necessity of matrix multiplication.

#### 1.1.1 Monte Carlo Power Method

The Monte Carlo power method involves a random sampling of the values in the matrix in such a way that a sequence converges to the principal eigenvalue. This method works for any matrix  $A$ .

We need to define a Markov chain on the *index* of the matrix  $A$ :  $1, \dots, n$ . Call this Markov chain  $\{k_n\}$  where a transition from  $k_n = \alpha$  to  $k_{n+1} = \beta$  depends on the magnitude of  $A_{\alpha\beta}$  in the following way

$$P(k_{n+1} = \beta | k_n = \alpha) = \frac{|A_{\alpha\beta}|}{\sum_{\beta} |A_{\alpha\beta}|}$$

with an arbitrary initial distribution generator  $\mathbf{h}$  so that

$$P(k_0 = \alpha) = \frac{|h_\alpha|}{\sum_\alpha |h_\alpha|}$$

Then we define a random sequence of variables  $W_n$  recursively:

$$W_0 = \frac{h_{k_0}}{p_{k_0}} \quad W_n = W_{n-1} \frac{A_{k_{n-1}k_n}}{p_{k_{n-1}k_n}}$$

It isn't hard to verify that

$$\lim_{n \rightarrow \infty} \frac{E[W_n \mathbf{f}_{k_n}]}{E[W_{n-1} \mathbf{f}_{k_{n-1}}]} = \lim_{n \rightarrow \infty} \frac{h^T A^n f}{h^T A^{n-1} f} = \lambda_{max}$$

for any  $\mathbf{f}$ . This method grows according to  $O(Nnm)$ , where  $N$  is the number of states in your Markov chain,  $n$  is the step number at when you terminate  $E[W_n f_{b_n}]$ , and  $m$  is the number of independent Monte Carlo paths that you use to construct  $E[W_n f_{b_n}]$ . However, in the contact process case, we can reduce this to  $O(Knm)$  where  $K$  is the number of nodes in the graph. The major drawback to this method is that it will only give you the approximate eigenvalue. In order to get the eigenvector, a lot more work is required especially for large matrices such as ours.

### 1.1.2 Other Methods

[2] is a good survey of other principal eigenvector estimation algorithms. [13] is a very recent pre-print of a stochastic method that is related but different from our method. The mirror descent method of [11] is also another alternative.

## 2 Background and Motivation

### 2.1 Contact Process

A contact process is a continuous-time Markov chain (CTMC)  $(X'_1, \dots, X'_n) \in \{0, 1\}^n$ , where  $t \geq 0$  is the time, with an associated connected graph  $(V, E)$  such that

- (i)  $|V| = n$
- (ii) Individual nodes transition from 1 to 0 at an exponential rate of 1
- (iii) Individual nodes transition from 0 to 1 at rate  $\lambda r$  where  $r$  is the fraction of neighbors that are in state 1

This CTMC has  $2^n$  states. The state  $(0, 0, \dots, 0)$  is an absorbing state and the remaining states are all transient.

This CTMC will eventually reach the absorbing state but physicists are interested in the ‘‘pseudo-equilibrium’’ behavior in the long time before absorption happens.

The hindrance of large state space can be alleviated in the rare cases where a compressed representation is possible, such as the case of a contact process on complete graph. In that case, we only need to know the total number of “on” states rather than the identities of all the “on” states.

## 2.2 Quasi-stationary Distribution

### 2.2.1 Discrete-Time Version

[3] proposed the concepts of quasi-stationary distribution and quasi-limiting distribution for the discrete-time Markov chains. Assume that 0 is the absorbing state and  $1, \dots, n$  are absorbing, we can write the Markov transition matrix as

$$P = \begin{bmatrix} 1 & 0 \\ \alpha & Q \end{bmatrix}$$

First we define the conditional transition probabilities

$$\begin{aligned} d_j^\pi(n) &= P(X_n = j | X_0 \sim \pi, X_1, \dots, X_{n-1} \neq 0) \\ &= \frac{\pi^t Q^{n-1} e_j}{\pi^t Q^{n-1} e} \end{aligned}$$

where  $\{e_i\}$  is the standard basis for  $R^n$  and  $e$  is the vector of all 1's. If there is a distribution  $\pi$  over the transient states such that  $d^\pi(n)$  is constant, then we call  $d^\pi$  the quasi-stationary distribution.

Under the assumption that the substochastic matrix  $Q$  is irreducible (not necessarily aperiodic), it is straightforward to see that the quasi-stationary distribution exists and is the unique solution to principal eigenvector problem

$$d^t Q = \rho d$$

by the Perron-Frobenius theorem.

Assuming  $Q$  is aperiodic and the condition that if  $|\rho_2| = |\rho_3|$ , we require the multiplicity of  $\rho_2$  to be no less than the multiplicity of  $\rho_3$ , we have that

$$d_j^\pi(n) \rightarrow d_j + O\left(n^k \frac{|\rho_2|}{\rho_1}\right)$$

Note that the rate of convergence depends on the ratio between the second eigenvalue and principle eigenvalue.

### 2.2.2 Continuous-Time

If we think about the transition rate matrix of a CTMC under similar setup (irreducibility and ergodicity), then it ([4]) can be said that

$$d_j^\pi(t) \rightarrow d_j + o(e^{t(\rho' - \rho_1)})$$

where  $d$  is the principal left-eigenvector of the rate matrix corresponding to the transient states with associated eigenvalue  $\rho_1$ . I.e.

$$d^t R = \rho_1 d^t$$

where  $R$  is the rate matrix of the CTMC.

### 2.3 Physicist's Heuristic

Under the setting of a continuous-time Markov chain with rate matrix  $R$  and absorbing state 0 (without loss of generality, we can combine all absorbing states into one state), if we define  $p_{ij}(t) = P(X_t = j | X_0 = i)$  and  $P_{is}(t) = 1 - p_{i0}(t)$ , then we have that quasi-stationary distribution  $d_j = \lim_{t \rightarrow \infty} \frac{p_{ij}(t)}{P_{is}(t)}$ . If we apply the Kolmogorov forward equation (known to physicists as the master equation), we get that

$$\frac{dp_{ij}(t)}{dt} = \sum_k p_{ik} R_{kj} \quad (1)$$

and

$$\frac{dP_{is}(t)}{dt} = \frac{d}{dt}(1 - p_{i0}(t)) = - \sum_k p_{ik} R_{k0} \quad (2)$$

Intuitively by the definition of  $d_j$ , we have that  $p_{ij}(t) \approx d_j P_{is}(t)$  in the quasi-stationary time window ( $t$  large enough). So we can apply this to the preceding two equations and get

$$\begin{aligned} d_j \frac{dP_{is}(t)}{dt} &= \sum_k d_k P_{is}(t) R_{kj} \\ \frac{dP_{is}(t)}{dt} &= - \sum_k d_k P_{is}(t) R_{k0} \end{aligned}$$

Combine the two and get

$$d_j \left( \sum_k d_k R_{k0} \right) + \sum_k d_k R_{kj} = 0$$

This gives us a non-linear equation for the equilibrium condition for the quasi-stationary distribution  $\mathbf{d}$ . We can think of this as the stationary point of the forward equation

$$\frac{d(d_j)}{dt} = \sum_k d_k R_{kj} + d_j \left( \sum_k d_k R_{k0} \right) \quad (3)$$

The first part of this equation is the standard Kolmogorov forward equation, while the second part redeposits the probability of hitting the absorbing states onto all the non-absorbing states according to the current distribution  $d_j$ .

This suggests the following algorithm

---

**Algorithm 1** Algorithm for estimating quasi-stationary distribution

---

- (i) Start the Markov chain in a non-absorbing state.
  - (ii) Simulate the Markov chain normally.
  - (iii) If the Markov chain hits the absorbing state, re-sample the starting position based on an empirical estimate of the quasi-stationary distribution up until that point and go to step 2. That is, we sample a non-absorbing state according to a weight proportional to the amount of time that such a state has been visited so far throughout the whole algorithm.
  - (iv) The samples after a large enough time window will be drawn approximately from the quasi-stationary distribution.
- 

For large enough time, the dynamics of the Markov chain will be governed by (3), which means we can obtain the quasi-stationary distribution by examining the empirical distribution after some large enough time.

### 3 Stochastic Approximation Analysis of the Algorithm

In this section, we will re-organize Algorithm 1 into a stochastic approximation algorithm. This will let us rigorously prove a convergence result and Central Limit Theorem for the algorithm.

#### 3.1 Formal Description of the Algorithm

We will now write down a precise description of the above Algorithm 1. Let our state space be the finite set  $S$  and  $T \subset S$  be the set of transient states. Let  $\mu_n$  be a probability measure over transient states.  $\mu_n$  will be the cumulative empirical distribution up until the  $n$ -th iteration. Let  $Q$  be the substochastic matrix over the transient states, and  $\{X_n^l\}_n$  be the  $l$ th Markov chain in the simulation, and  $\tau^l = \min\{k \geq 0 | X_k^l \notin T\}$  (the hitting time of the absorbing state), we can write our algorithm as

$$\mu_{n+1}(x) = \frac{\left( \sum_{l=0}^n \tau^l \right) \mu_n(x) + \left( \sum_{k=0}^{\tau^{n+1}-1} I(X_k^{n+1} = x | X_0^{n+1} \sim \mu_n) \right)}{\sum_{l=0}^{n+1} \tau^l} \quad \forall x \in T$$

for any arbitrary initial distribution  $\mu_0$ .

Robbins-Monro, or stochastic approximation algorithms [12, 10], have the form

$$\mu_{n+1} = \mu_n + \alpha_n Y(\mu_n)$$

where

$$\sum_n \alpha_n = \infty \quad \sum_n \alpha_n^2 < \infty \quad \alpha_n \geq 0 \quad \alpha_n \rightarrow 0$$

and  $Y(\cdot)$  is a collection of vector-valued random variables for each possible point in the state-space. Note that over the years, the form of the Robbins-Monro algorithm has been extended. The form here is the classical version.

Under certain conditions, which will be discussed rigorously in [11],  $\mu_n$  converges to root of the function  $g(\mu) \triangleq E[Y(\mu)]$ . We will transform  $\mu_n$  into stochastic approximation

$$\mu_{n+1}(x) = \mu_n(x) + \left( \frac{1}{n+1} \right) \left( \frac{\sum_{l=0}^{\tau^{(n+1)}-1} \left( I(X_l^{(n+1)} = x) - \mu_n(x) \right)}{\frac{1}{n+1} \sum_{j=0}^{n+1} \tau^{(j)}} \right)$$

where

$$Y(\mu) = \frac{\sum_{l=0}^{\tau-1} (I(X_l = x | X_0 \sim \mu) - \mu(x))}{\frac{1}{n+1} \sum_{j=0}^{n+1} \tau^{(j)}}$$

The denominator is problematic because it depends on the whole history of  $\mu_n$  and not just on the present state. To solve this, we artificially consider another state  $T_n$  in the following way.

Stochastic approximation scheme for the main algorithm

$$T_{n+1} = T_n + \frac{1}{n+2} (\tau^{(n+1)} - T_n) \Rightarrow T_n = \frac{1}{n+1} \sum_{j=0}^n \tau^{(j)}$$

$$\mu_{n+1}(x) = \mu_n(x) + \tag{4}$$

$$\left( \frac{1}{n+1} \right) \left( \frac{\sum_{l=0}^{\tau^{(n+1)}-1} \left( I(X_l^{(n+1)} = x | X_0^{(n+1)} \sim \mu_n) - \mu_n(x) \right)}{T_n + \frac{\tau^{(n+1)}}{n+1}} \right) \tag{5}$$

we can therefore define

$$Y_n(\mu_n, T_n)(x) \triangleq \frac{\sum_{l=0}^{\tau^{(n+1)}-1} \left( I(X_l^{(n+1)} = x) - \mu_n(x) \right)}{T_n + \frac{\tau^{(n+1)}}{n+1}}$$

$$Z_n(\mu_n, T_n) \triangleq (\tau^{(n+1)} - T_n)$$

So now we have a stochastic approximation path  $(\mu_n, T_n)$ , where the control parameters are  $(\mu, T)$ , that fits into the Robbins-Monro scheme above.

**Remark 1** Please note that the iterates  $\mu_n$  are constrained in  $H \triangleq \{\mathbf{x} \in \mathbb{R}_+^n \mid \sum x_i = 1\}$ , the  $(n-1)$ -dimensional simplex.

We can also define a similar algorithm for the continuous-time Markov chain by keeping track of the amount of time a Markov chain spends in each transient state. This is given in Sect. 5.

We can summarize the conditions for our algorithm in the following theorem taken from Blanchet, Glynn, and Zheng (2012):

**Theorem 1** Given an irreducible absorbing Markov chain over a finite state space  $S$  of cardinality  $d$ , let

- (i) Matrix  $Q$  denoting the transition rates over the non-absorbing states
- (ii) Let  $\mu_0$  be a probability vector over the non-absorbing states
- (iii) Let  $T_0 \in \mathbb{R}^+$

Then there exists a unique quasi-stationary distribution  $\mu$  satisfying the equations

$$\begin{aligned}\mu^t Q &= \lambda \mu \\ \mu^t \mathbf{1} &= 1 \\ \mu &\geq 0\end{aligned}$$

and the Algorithm 1 converges to the point  $(\mu, \frac{1}{1-\lambda})$  with probability 1.

Furthermore, if  $\lambda_{PV}$  is the principal eigenvalue of  $Q$  and  $\lambda_{NPV}$  are the other eigenvalues and they satisfy

$$\operatorname{Re} \left( \frac{1}{1 - \lambda_{NPV}} \right) < \frac{1}{2} \left( \frac{1}{1 - \lambda_{PV}} \right) \quad \forall \lambda_{NPV} \text{ non-principal eigenvalues}$$

Then

$$\sqrt{n}(\mu_n - \mu) \rightarrow^d N(0, V)$$

for some matrix  $V$ .

### 3.2 Sketch of Proof of Convergence

Our proof in [11] rests on the use of the ODE method [10] where we are required to examine the asymptotics of the coupled dynamical system

$$\dot{\mu}(t) = E_{\mu(t), T(t)} \left[ \frac{\sum_{l=0}^{\tau-1} (I(X_l = \cdot | X_0 \sim \mu)) - \tau \mu(x)}{T(t)} \right]$$

$$\text{(Define)} \quad A \triangleq (I - Q)^{-1} = \frac{1}{T} [\mu^t(t)A - (\mu^t A \mathbf{1})\mu^t(t)]$$

$$\begin{aligned}\dot{T}(t) &= E_\mu[\tau] - T \\ &= \mu^t(t)(I - Q)^{-1}\mathbf{1} - T(t)\end{aligned}$$

where  $\mu \in R^n$  and  $T \in R^+$ .

In [11], we were able to show that for a given initial position in the probability simplex, the solution to the above dynamical system exists and converges to its stationary point which is the unique point that satisfies

$$\begin{aligned}\mu^T Q &= \rho \mu^T \\ \sum \mu_i &= 1 \\ \mu_i &\geq 0\end{aligned}$$

and  $\rho = 1 - \frac{1}{E_\mu(\tau)}$ .

By Theorem 4.2.1 from [10], we can conclude that  $\mu_n$  converges to the quasi-stationary distribution for all initial configurations  $(\mu_0, T_0)$ .

By Theorem 10.2.1 from [10], we conclude that a Central Limit Theorem exists as long as the Jacobian of the ODE vector field has spectral radius less than  $-0.5$ . This is equivalent to requiring that

$$Re\left(\frac{1}{1 - \lambda_{NPV}}\right) < \frac{1}{2} \left(\frac{1}{1 - \lambda_{PV}}\right) \quad \forall \lambda_{NPV} \text{ non-principal eigenvalues} \quad (6)$$

where the  $\lambda$ 's are the eigenvalues of the  $Q$  matrix.

## 4 Variations on the Existing Algorithm with Improved Rate of Convergence

One interesting question to ask is what happens when the sufficient conditions for Central Limit Theorem is not met. We will study a simple example consisting of two states.

### 4.1 Counter Example to CLT

Imagine we have a Markov chain with three states  $\{0, 1, 2\}$  and transition matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & \frac{1-\varepsilon}{2} & \frac{1-\varepsilon}{2} \\ \varepsilon & \frac{1-\varepsilon}{2} & \frac{1-\varepsilon}{2} \end{bmatrix}$$



Obviously the state  $\{0\}$  is the absorbing state. In this setup, because of symmetry, our Algorithm [1](#) reduces to

- (i) With probability  $\frac{1-\varepsilon}{2}$  sample either the state 1 or 2 (without knowing the previous state, this is ok by symmetry)
- (ii) With probability  $\varepsilon$ , sample from either 1 or 2 according to the empirical distribution up until this point.

We recognize this as a self-interacting Markov chain.

A self-interacting Markov chain [\[7\]](#) is a stochastic process  $\{X_n\}$  such that

$$P(X_{n+1} \in dx | \mathcal{F}_n) = \Phi(S_n)(dx)$$

where  $\Phi$  is a function that transforms one measure into another measure and  $S_n$  is the empirical measure generated by  $\{X_k\}_{k=0}^n$ .

Then our quasi-stationary algorithm reduces to the empirical process of a SIMC  $X_n$  governed by

$$P(X_{n+1} = dz | \mathcal{F}_n) = \int K(x, dz) dS_n(dx)$$

where the kernel is given by

$$K(x, dz) = \varepsilon \delta_x(dz) + \left(\frac{1-\varepsilon}{2}\right) [\delta_1(dz) + \delta_2(dz)]$$

The sufficient condition for CLT [\(6\)](#) or this problem translates to requiring  $\varepsilon < 0.5$ .

When the CLT is violated however, [\[7\]](#) states that over a very general class of bounded and measurable functions  $f$

$$E[(S_n(f) - \bar{S}_n(f))^2] = \Theta\left(\frac{1}{n^{2(1-\varepsilon)}}\right)$$

where  $S_n(f) = \int f(x) dS_n(x)$ ,  $\bar{S}_n(f) = E[S_n(f)]$ . Although this doesn't technically contradict with the existence of a  $\sqrt{n}$ -CLT, it does suggest that the scaling sequence is  $n^{1-\varepsilon}$  instead of  $\sqrt{n}$ .

## 4.2 The Parallel Algorithm

There is a variant of the algorithm that can offer significant practical benefits. Imagine that at each iteration, instead of there being one run of the Markov chain until absorption, we have  $M$  independent runs. Such that

$$\begin{aligned} \mu_{n+1}(x) &= \frac{\mu_n(x) \left( \sum_{l=0}^n \sum_{m=1}^M \tau^{l,m} \right) + \sum_{m=1}^M \left[ \sum_{k=0}^{\tau^{n+1,m}-1} I(X_k^{n+1,m} = x | X_0^{n+1,m} \sim \mu_n) \right]}{\sum_{l=0}^{n+1} \sum_{m=1}^M \tau^{l,m}} \\ &= \mu_n(x) + \frac{1}{n+1} \frac{\sum_{m=1}^M \left[ \sum_{k=0}^{\tau^{n+1,m}-1} \left( I(X_k^{n+1,m} = x | X_0^{n+1,m} \sim \mu_n) \right) - \tau^{n+1,m} \mu_n(x) \right]}{\frac{1}{n+1} \sum_{l=0}^{n+1} \sum_{m=1}^M \tau^{l,m}} \end{aligned}$$

Again we have to include an extra dimension

$$\mu_{n+1} = \mu_n + \frac{1}{n+1} \left( \frac{\sum_{m=1}^M \left[ \sum_{k=0}^{\tau^{n+1,m}-1} \left( I(X_k^{n+1,m} = \cdot | X_0^{n+1,m} \sim \mu_n) \right) - \tau^{n+1,m} \mu_n \right]}{T_n + \frac{1}{n+1} \sum_{m=1}^M \tau^{n+1,m}} \right)$$

$$T_{n+1} = T_n + \frac{1}{n+2} \left( \sum_{m=1}^M \tau^{m,n+1} - T_n \right) \Rightarrow T_n = \frac{1}{n+1} \sum_{j=0}^n \sum_{m=1}^M \tau^{m,j}$$

After some derivation, we obtain the dynamical system

$$\dot{\mu}(t) = \frac{M}{T(t)} (\mu^t (I - Q)^{-1} - (\mu^t (I - Q)^{-1} \mathbf{1}) \mu^t)$$

$$\dot{T}(t) = M \mu^t (I - Q)^{-1} \mathbf{1} - T$$

Very similarly, we know that

$$\mu(t) \rightarrow \bar{\mu}$$

$$T(t) \rightarrow \frac{M}{1 - \bar{\lambda}}$$

If we let  $g^\mu$  and  $g^T$  denote the dynamical system's components, then we obtain the Jacobian

$$\nabla_\mu g^\mu = \frac{M}{T} \left( (I - Q)^{-1} - (I - Q)^{-1} \mathbf{1} \mu^T - \frac{1}{1 - \bar{\lambda}} I \right)$$

$$\nabla_T g^\mu = -\frac{M}{T^2} (\mu^t (I - Q)^{-1} - (\mu^t (I - Q)^{-1} \mathbf{1}) \mu^t)$$

$$\nabla_\mu g^T = M (I - Q)^{-1} \mathbf{1}$$

$$\nabla_T g^T = -1$$

So the condition for which the Central Limit Theorem remains the same:

$$\text{Re} \left( \frac{1}{1 - \lambda_{NPV}} \right) < \left( \frac{1}{2} \right) \frac{1}{1 - \lambda_{PV}} \quad \forall \lambda_{NPV} \text{ non principal eigenvalues}$$

where  $\lambda_{PV}$  is the principal eigenvalue of  $Q$  and  $\lambda_{NPV}$  is the non-principal eigenvalue of  $Q$ .

Although the Central Limit Theorem does not always hold, the variance of the stochastic approximation noise is lower with bigger  $M$ . This means that if we have enough independent Markov chain iterations across different processors, the algorithm would perform better. See Section 5 for empirical performance.

## 5 Continuous-Time Markov Chains

### 5.1 Formulation and Convergence

So far, the exposition has assumed that the Markov chain of interest is a discrete-time process. It is straightforward to adapt our method for continuous-time processes (such as the contact process). If we denote the transition rate matrix of the CTMC in the following block form

$$T = \begin{bmatrix} 0 & 0 \\ N & Q \end{bmatrix}$$

then we can write the algorithm as

$$\begin{aligned} \mu_{n+1}(x) &= \frac{\mu_n(x) \left( \sum_{l=0}^n \sum_{m=1}^M \tau^{l,m} \right) + \sum_{m=1}^M \left[ \int_0^{\tau^{n+1,m}} I(X^{n+1,m}(s) = x | X_0^{n+1,m} \sim \mu_n) ds \right]}{\sum_{l=0}^{n+1} \sum_{m=1}^M \tau^{l,m}} \\ &= \mu_n(x) + \frac{1}{n+1} \frac{\sum_{m=1}^M \left[ \int_0^{\tau^{n+1,m}} \left( I(X^{n+1,m}(s) = x | X_0^{n+1,m} \sim \mu_n) \right) ds - \tau^{n+1,m} \mu_n(x) \right]}{\frac{1}{n+1} \sum_{l=0}^{n+1} \sum_{m=1}^M \tau^{l,m}} \end{aligned}$$

By a similar approach as the discrete-time case, we deduce the related dynamical system

$$\begin{aligned} \dot{\mu}(t) &= -\frac{M}{T(t)} \left( \mu^t Q^{-1} - (\mu^t Q^{-1} \mathbf{1}) \mu^t \right) \\ \dot{T}(t) &= -M \mu^t Q^{-1} \mathbf{1} - T \end{aligned}$$

It is straightforward to adapt the Perron-Frobenius theorem to transition rate matrices such as  $Q$  by decomposing  $Q = A - bI$  where  $A$  is an irreducible matrix. We know the existence of a principal eigenvector of positive entries  $\bar{\mu}$  (with eigenvalue smaller than 0) such that

$$\bar{\mu}^t Q = \bar{\lambda} \bar{\mu}^t$$

We can easily check that the stationary point, and with more work the limit point of the dynamical system satisfies

$$\begin{aligned} \bar{\mu}^t Q &= \frac{1}{\bar{\mu}^t Q^{-1} \mathbf{1}} \bar{\mu}^t = \bar{\mu}^t \\ \bar{T} &= -M \bar{\mu}^t Q^{-1} \mathbf{1} = -M \frac{1}{\bar{\lambda}} \end{aligned}$$

Hence we have proven that the CTMC version of the algorithm converges to the quasi-stationary distribution.

## 5.2 Rate of Convergence

The Jacobian of the dynamical system is given by

$$\begin{aligned}\nabla_{\mu} g^{\mu} &= -\frac{M}{T} (Q^{-1} - Q^{-1} \mathbf{1} \mu^T - (\mu^T Q^{-1} \mathbf{1}) I) \\ \nabla_T g^{\mu} &= \frac{M}{T^2} (\mu^T Q^{-1} - (\mu^T Q^{-1} \mathbf{1}) \mu^T) \\ \nabla_{\mu} g^T &= -M Q^{-1} \mathbf{1} \\ \nabla_T g^T &= -1\end{aligned}$$

When evaluated at the stationary point  $(\bar{\mu}, \bar{T})$ , we get the matrix

$$\begin{bmatrix} -\bar{\lambda} \left( Q^{-1} - Q^{-1} \mathbf{1} \bar{\mu}^T - \frac{1}{\bar{\lambda}} I \right) & -M Q^{-1} \mathbf{1} \\ \mathbf{0} & -1 \end{bmatrix}$$

If  $\lambda_Q$  is any non-principal eigenvalue of  $Q$ , then the sufficient condition for CLT becomes

$$2\lambda_{PV} > \text{Re}(\lambda_Q)$$

## 5.3 Uniformization

Because these CTMC have finite state space, we can form the associated uniformized Markov chain. Let  $Q$  be the transition rate matrix of the non-absorbing states and let  $\nu = \max_i (-q_{ii})$ , we can form a discrete-time transition matrix

$$\tilde{Q} = I + \frac{1}{\nu} Q$$

It is straightforward to verify that any principal left-eigenvector to  $Q$  is also a principal left-eigenvector to  $\tilde{Q}$ . Hence we apply the discrete-time algorithm to this DTMC.

## 6 Numerical Experiments

### 6.1 Loopy Markov Chain

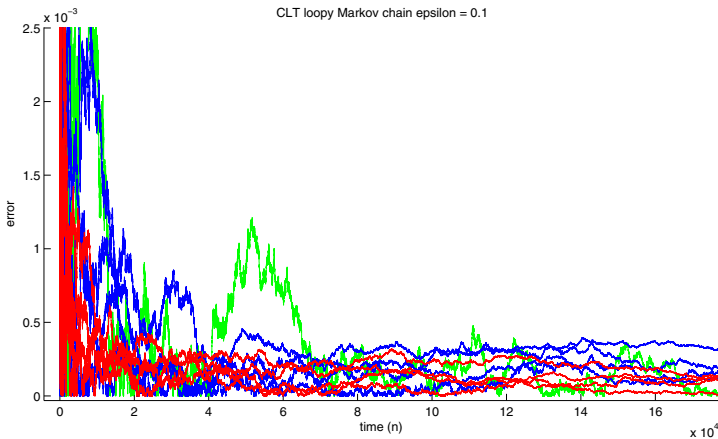
Let's consider the loopy Markov chain given by the full stochastic matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & \frac{1-\varepsilon}{2} & \frac{1-\varepsilon}{2} \\ \varepsilon & \frac{1-\varepsilon}{2} & \frac{1-\varepsilon}{2} \end{bmatrix}$$

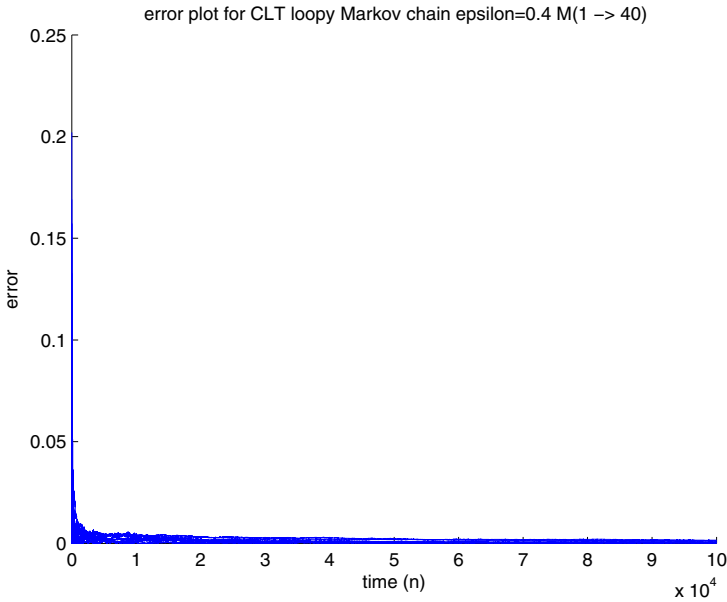
the eigenvalues of the sub-stochastic matrix are  $1 - \varepsilon$  and 0. Hence the sufficient condition for Central Limit Theorem to hold is requiring  $\varepsilon < 0.5$ . A series of numerical experiments were performed for different values of  $\varepsilon$  where the L2 error is plotted against time. The observation is summarized in the following table.

**Table 1** This table summarizes the behavior of the loopy Markov chain for various  $\varepsilon$

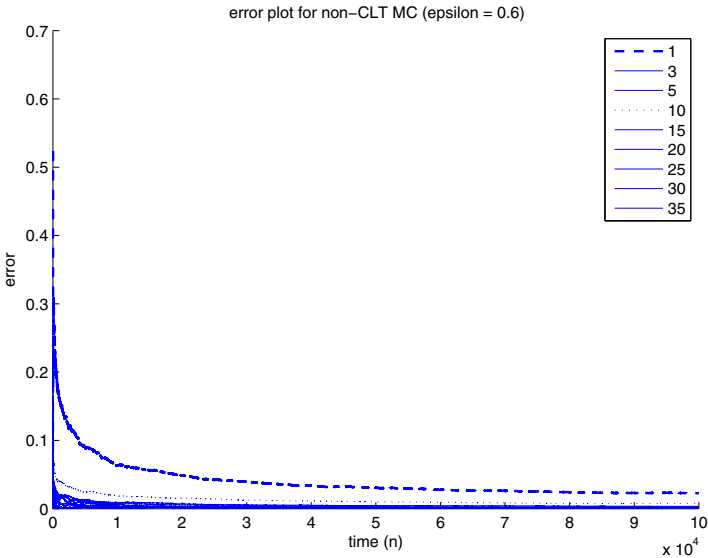
$\varepsilon$	CLT Satisfied?	Observation	Figure (in appendix)
0.1	yes	No difference between the performance of different $M$ 's.	A.1
0.4	yes (borderline)	No difference between the performance of different $M$ 's.	A.2
0.6	no (borderline)	Noticeable, but relatively medium difference between small $M$ and larger $M$ . Observed critical $M=2$ . Anomalous large error for the $M=10$ run.	A.2
0.98	no	Huge difference between the simulation with small $M$ and larger $M$ . However, some of the simulations with very large $M$ begin to show larger errors than the simulation with medium $M$ 's.	A.4



**Fig. 1** This figure is the time vs. error plot of the main algorithm ran on a loopy Markov chain with eigenvalues well within the CLT regime ( $\varepsilon = 0.1 < 0.5$ ). Notice the scale of the y-axis. The colors of the different lines represent different runs with different  $M$ 's. In this regime which satisfies the CLT for all  $M$ , increasing  $M$  does not improve the rate of convergence.



**Fig. 2** This figure is the time vs. error plot of the main algorithm run on a loopy Markov chain with eigenvalues just within the CLT regime ( $\epsilon = 0.4 < 0.5$ ). Just like the previous figure, this figure shows that increasing  $M$  does not improve the rate of convergence.

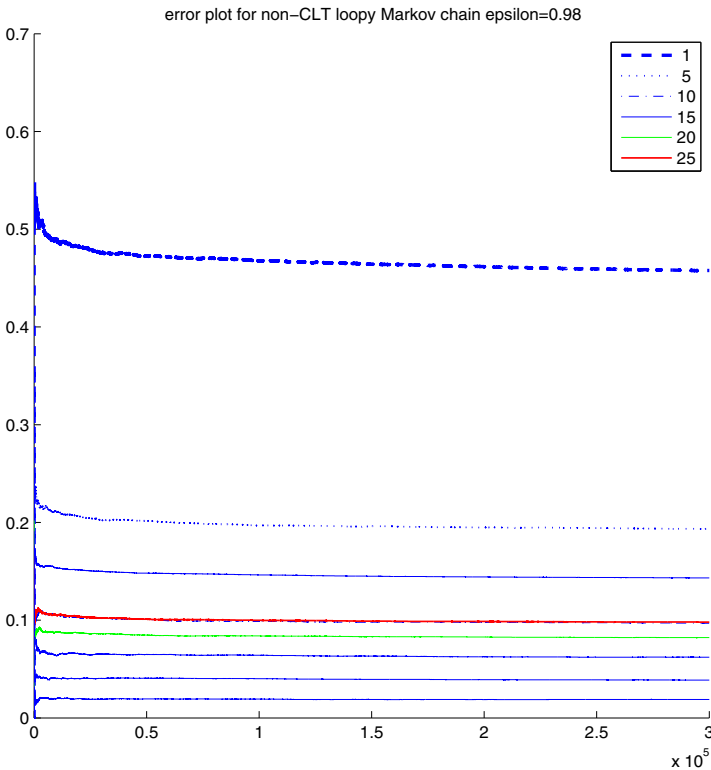


**Fig. 3** This is the time vs. error plot of the main algorithm run on a Markov chain with eigenvalues just outside of the CLT regime ( $\epsilon = 0.6 > 0.5$ ). As you can see, there is a noticeable difference between the  $M = 1$  simulation and other  $M$  simulations. However, there is an anomalous run for  $M = 10$ . It is probably due to the inherent large variance of the error.

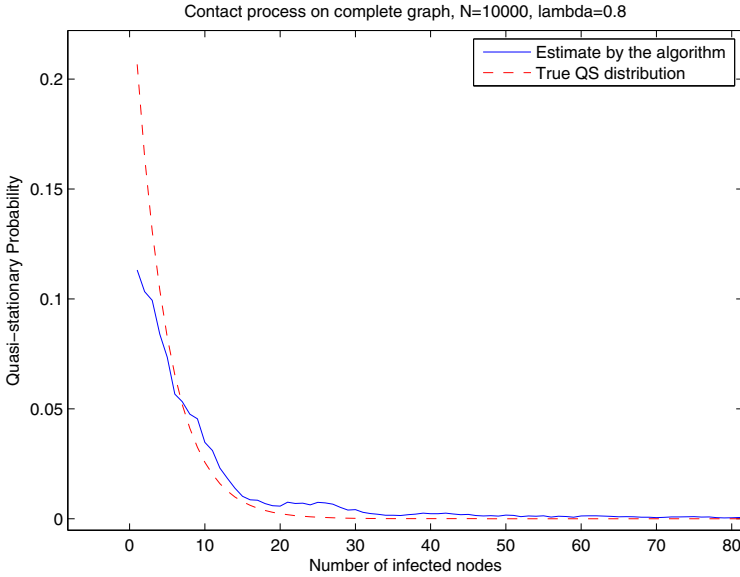
## 6.2 Contact Process on Complete Graph

### 6.2.1 Low Infection Rate

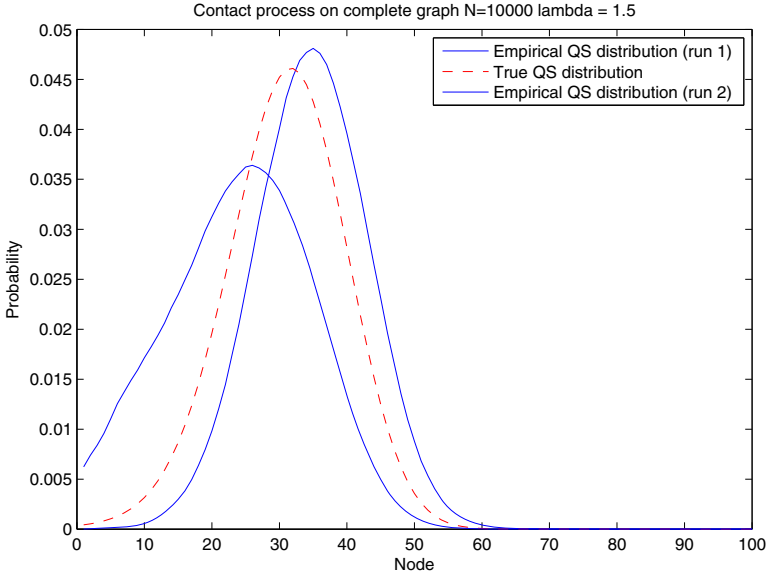
Here we apply the algorithm to a special case of the contact process. This is the contact process on a complete graph. This simple case allows the process to be only represented by the number of infected nodes. We picked 10000 nodes and an infection rate of 0.8. The algorithm was run for 1000 iterations. See Fig. 5 for the plot of the estimated distribution vs the true distribution.



**Fig. 4** This is the time vs. error plot of the main algorithm run on a Markov chain with eigenvalues just outside of the CLT regime ( $\epsilon = 0.98 \gg 0.5$ ). As you can see, there are huge differences between the  $M = 1, 5$  simulation and other  $M$  simulations. However, there are anomalous runs for  $M = 20, 25$ . They are probably due to the inherent large variance of the error.



**Fig. 5** This is the time vs. probability plot of the the continuous-time version of the algorithm applied to the contact process on complete graph with 10000 nodes and an infection rate of 0.8.



**Fig. 6** This is the time vs. probability plot of he continuous-time version of the algorithm applied to the contact process on complete graph with 10000 nodes and a high infection rate of 1.5.



### 6.2.2 High Infection Rate

If the infection rate is changed to 1.5, then each iteration of the algorithm would take an extreme long time due to the time it takes to hit the absorbing state. Hence, we uniformized the continuous-time chain to get a discrete-time transition matrix  $Q$ . Instead of applying the algorithm to  $Q$ , we can apply the algorithm to  $0.99 \times Q$  in order to shorten each tour. The algorithm showed high variability on the two different runs. See Fig. 6 for the plot of the estimated distribution vs the true distribution.

## 7 Discussion and Conclusion

In summary, we have given a rigorous foundation to the algorithm of [5] by recognizing it as a stochastic approximation algorithm. In doing so, we were able to prove its law of large number and fluid limits. A slightly improved algorithm is also proposed and this algorithm significantly improves rate of convergence for some cases.

There also exists a class of projection-based stochastic approximation algorithms  $\theta_{n+1} = \Pi[\theta_n + \varepsilon_n Y_n]$  that can be applied to our algorithm. Namely, we can discard the ‘‘T’’ dimension in our algorithm and replace the normalizing denominator by a projection operator. Unfortunately, this algorithm works very poorly in practice.

We have tested our algorithm on countable state space processes such as the M/M/1/ $\infty$  queue with success. Proving the convergence of this algorithm in this setting is currently an open problem.

**Acknowledgements.** Support from the NSF foundation through the grants CMMI-0846816 and CMMI-1069064 is gratefully acknowledged.

## References

1. Blanchet, J., Glynn, P., Zheng, S.: Theoretical analysis of a stochastic approximation algorithm for computing quasi-stationary distributions. Technical report, Columbia University (2012)
2. Chatterjee, C.: Adaptive algorithms for first principal eigenvector computation. *Neural Networks* 18, 145–159 (2005)
3. Darroch, J.N., Seneta, E.: On quasi-stationary distributions in absorbing discrete-time finite markov chains. *Journal of Applied Probability* 2(1), 88–100 (1965)
4. Darroch, J.N., Seneta, E.: On quasi-stationary distributions in absorbing continuous-time finite markov chains. *Journal of Applied Probability* 4, 192–196 (1967)
5. de Oliveira, M.M., Dickman, R.: How to simulate the quasistationary state. *Physical Review E* 71 (2005)
6. de Oliveira, M.M., Dickman, R.: Quasi-stationary simulation: the subcritical contact process. *Brazilian Journal of Physics* (2006)
7. del Moral, P., Miclo, L.: Self-interacting markov chains. *Stochastic Analysis and Applications* 24, 615–660 (2006)
8. Dickman, R., Vidigal, R.: Quasi-stationary distributions for stochastic processes with an absorbing state. *Journal of Physics A* 35, 1147–1166 (2002)

9. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press (1996)
10. Kushner, H.J., Yin, G.: Stochastic approximation and recursive algorithms and applications, vol. 35. Springer (2003)
11. Nazin, A.: Estimating the principal eigenvector of a stochastic matrix: Mirror descent algorithms via game approach with application to pagerank problem. In: 2010 49th IEEE Conference on Decision and Control, pp. 792–797 (2010)
12. Robbins, H., Monro, S.: A stochastic approximation method. *The Annals of Mathematical Statistics* 22, 400–407 (1951)
13. Whiteley, N., Kantas, N.: A particle method for approximating principal eigen-functions and related quantities. arXiv preprint

**Part II**  
**Genetic Programming**

# Locality in Continuous Fitness-Valued Cases and Genetic Programming Difficulty

Edgar Galvan, Leonardo Trujillo, James McDermott, and Ahmed Kattan

**Abstract.** It is commonly accepted that a mapping is local if it preserves neighbourhood. In Evolutionary Computation, locality is generally described as the property that neighbouring genotypes correspond to neighbouring phenotypes. Locality has been classified in one of two categories: high and low locality. It is said that a representation has high locality if most genotypic neighbours correspond to phenotypic neighbours. The opposite is true for a representation that has low locality. It is argued that a representation with high locality performs better in evolutionary search compared to a representation that has low locality. In this work, we explore, for the first time, a study on Genetic Programming (GP) locality in continuous fitness-valued cases. For this, we extended the original definition of locality (first defined and used in Genetic Algorithms using bitstrings) from genotype-phenotype mapping to the genotype-fitness mapping. Then, we defined three possible variants of locality in GP regarding neighbourhood. The experimental tests presented here use a set of symbolic regression problems, two different encoding and two different mutation operators. We show how locality can be studied in this type of scenarios (continuous fitness-valued cases) and that locality can successfully be used as a performance prediction tool.

---

Edgar Galvan

Distributed Systems Group, School of Computer Science and Statistics,  
Trinity College Dublin, Ireland

e-mail: [edgar.galvan@scss.tcd.ie](mailto:edgar.galvan@scss.tcd.ie)

Leonardo Trujillo

Departamento de Ingeniería Eléctrica y Electrónica,  
Instituto Tecnológico de Tijuana, Tijuana, BC, México

e-mail: [leonardo.trujillo.ttl@gmail.com](mailto:leonardo.trujillo.ttl@gmail.com)

James McDermott

EvoDesignOpt, MIT CSAIL, Cambridge, MA, USA

e-mail: [jamesmichaelmcdermott@gmail.com](mailto:jamesmichaelmcdermott@gmail.com)

Ahmed Kattan

Computer Science Department, Loughborough University, United Kingdom

e-mail: [A.J.Kattan@lboro.ac.uk](mailto:A.J.Kattan@lboro.ac.uk)

## 1 Introduction

Over the last years, researchers in the Evolutionary Computation (EC) community have been trying to estimate problem difficulty and landscape's structures (e.g., [5, 6, 8, 10, 13, 14, 17, 18]). In particular, researchers have focused their attention on the use of Genetic Algorithms [9] for their studies. In this work, we make an effort to understand problem difficulty by using a more complex representation and use Genetic Programming (GP) [11], where there are some interesting works that have shed some light on this research area [5, 17, 18].

One element that underlies many of these approaches is the well-known notion of fitness landscape, originally described in [20]. Over the years, researchers have defined fitness landscapes in slightly different ways. All of them have in common the use of three main elements: search space  $x$ , neighbourhood mapping  $\chi$  and fitness function  $f$ . More formally, a fitness landscape, as specified in [15], is normally defined as a triplet  $(x, \chi, f)$ : (a) a set  $x$  of configurations, (b) a notion  $\chi$  of neighbourhood, distance or accessibility on  $x$ , and finally, (c) a fitness function  $f$ . The graphical representation of this, whenever possible, can give an indication about the difficulty of the problem.

How an algorithm explores and exploits a landscape is a key element of evolutionary search. Rothlauf [14] has described and analysed the importance of locality in performing an effective evolutionary search of landscapes.

In EC, locality refers to how well neighbouring genotypes correspond to neighbouring phenotypes, and is useful as an indicator of problem difficulty. Similarly, the principle of strong causality states that for successful search, a small genotypic change should result in a small fitness change [2].

In his research, Rothlauf distinguished two forms of locality, low and high. A representation has high locality if most neighbouring genotypes correspond to neighbouring phenotypes, that is, small genotypic changes result in small phenotypic changes. On the other hand, a representation has low locality if many neighbouring genotypes do not correspond to neighbouring phenotypes. According to Rothlauf, a representation that has high locality is better compared to a representation with low locality. It should be noticed that a quantitative locality measure denoting high locality is close to 0, whereas the opposite is true for low locality. In his original studies, Rothlauf used GAs with bitstrings to conduct his experiments [14]. To our knowledge, there are few explicit studies on locality using the typical GP representation (i.e., tree-like structures).

The goal of this paper then, is to shed some light on the degree of locality present in GP. In particular in continuous-valued fitness. It is worth mentioning that there are some works on GP locality [4, 5]. However, to the best of our knowledge, none of them have studied this scenario in detail. Thus, in this study we cover the following:

- We extend the notion of genotype-phenotype locality to the genotype-fitness case for continuous-values fitness. For this purpose we treat two individuals as neighbours in the genotype space if they are separated by a single mutation operation.

- We consider three different definitions of locality to study which of them gives the best prediction of problem difficulty,
- We consider only a mutation based GP (two different mutation operators), and
- Finally, we use two different encodings on five different problems (i.e., Real-Valued Symbolic Regression problems) and compare the results against the three definitions of locality.

This paper is organised as follows. In the following section a more detailed explanation on locality is provided. Section 3 presents previous work on performance prediction. Section 4 presents the experimental setup. In Section 6 we present and discuss our findings. Finally, in Section 6 we draw some conclusions.

## 2 Locality

Understanding how well neighbouring genotypes correspond to neighbouring phenotypes is a key element in understanding evolutionary search [14]. In the abstract sense, a mapping has *locality* if neighbourhood is preserved under that mapping<sup>1</sup>. In EC this generally refers to the mapping from genotype to phenotype. This is a topic worthy of study because if neighbourhood is not preserved, then the algorithm’s attempts to exploit the information provided by an individual’s fitness will be misled when the individual’s neighbours turn out to be very different.

Rothlauf gives a quantitative definition of locality: “the locality  $d_m$  of a representation can be defined as

$$d_m = \sum_{d^g(x,y)=d_{\min}^g} |d^p(x,y) - d_{\min}^p|$$

where  $d^p(x,y)$  is the phenotypic distance between the phenotypes  $x$  and  $y$ ,  $d^g(x,y)$  is the genotypic distance between the corresponding genotypes, and  $d_{\min}^p$  resp.  $d_{\min}^g$  is the minimum distance between two (neighbouring) phenotypes, resp. genotypes” [14, p. 77; notation changed slightly]. Locality is thus seen as a continuous property rather than a binary one. It should be noticed, however, that this measure has been (almost) exclusively used in discrete scenarios [14].

The point of this definition is that it provides a single quantity which gives an indication of the behaviour of the genotype-phenotype mapping which can be compared between different representations.

It is worth pointing out that while Rothlauf gives a quantitative definition of locality, as expressed above ( $d_m$ ), this in fact measures phenotypic divergence. Once a value is obtained by this measure, then we can talk about high and low locality. To avoid confusion regarding the type of locality present in a representation, it is necessary to highlight the following: when the phenotypic divergence  $d_m$  is low, we

<sup>1</sup> The term *locality* has also been used in an unrelated context, to refer to the quasi-geographical distribution of an EC population [3].

are in presence of high locality. On the other hand, when the phenotypic divergence  $d_m$  is high, we are in the presence of “non-locality” (originally called “low-locality” [14]). We have decided to re-label the latter term to avoid confusion. These two terms (high and non-locality) will be used throughout the paper.

Rothlauf stated that a representation that has high locality will be more efficient at evolutionary search. If a representation has this type of locality (i.e., neighbouring genotypes correspond to neighbouring phenotypes) then performance is good. This, however, changes when a representation has non-locality, i.e., non-locality will translate to a poor evolutionary performance.

To use locality as a measure of problem difficulty in GP, it is necessary to extend the standard definition of genotype-phenotype locality given by Rothlauf [14] to the genotype-fitness mapping [4, 5]. This is because, it is normally accepted that in tree-structured GP there is no explicit genotype-phenotype mapping, so we can say that there are no explicit phenotypes distinct from genotypes.

In [4, 5], Galvan et al. extended the typical definition of locality from the genotype-phenotype to the genotype-fitness mapping. In their work, the authors were able to show how to predict performance on discrete valued-fitness cases. In this work, as mentioned previously, we want to extend this work to continuous-valued fitness. This is presented in the following section.

## 2.1 Extending the Definition of Locality to the Continuous Valued-Fitness

In [4, 5], the authors introduced, for the first time, the definition of locality in GP showing how locality correctly predicted performance only in *discrete* valued-fitness cases. Thus, our intention is to extend GP locality to *continuous* valued-fitness cases. For this purpose, we generalise the three definitions of locality originally presented in [4, 5]. These three definitions can be summarised as follows:

- Regard two individuals as fitness-neighbours if the difference of their fitness values is 1, and regard fitness-neutral mutations as non-local. This leads to the following definition for “non-locality” which we call Def<sub>0</sub>:

$$d_m = \frac{\sum_{i=1}^N |fd(x_i, m(x_i)) - fd_{\min}|}{N} \quad (1)$$

where  $fd(x_i, m(x_i)) = |f(x_i) - f(m(x_i))|$  is the fitness distance between a randomly-sampled individual  $x_i$  and the mutated individual  $m(x_i)$ ,  $fd_{\min} = 1$  is the minimum fitness distance between two individuals, and  $N$  is the sample size.

- Def<sub>0</sub>, however, treats a fitness-neutral mutation as being just as bad for locality as a mutation causing a fitness divergence of two fitness units (assuming integer-valued fitness). It might be preferable to redefine the minimum distance in the fitness space as zero, giving the same locality definition as above but with  $fd_{\min} = 0$ . We called this Def<sub>1</sub>.

- Finally, it might be better to treat only true divergence of fitness as indicating poor locality. Therefore we might say that fitness divergence occurs only when the fitness distance between the pair of individuals is 2 or greater: otherwise the individuals are regarded as neighbours in the fitness space. This leads to the following definition, which we will call Def<sub>2</sub>.

$$d_m = \frac{\sum_{i=1:fd(x_i,m(x_i)) \geq 2}^N fd(x_i,m(x_i))}{N} \quad (2)$$

As mentioned previously, the work developed in [4, 5] considers the study of GP locality in discrete-valued cases. In the following paragraphs, we show how it is possible to extend these definition to continuous-valued fitness. To this end, we consider the same definitions previously summarised. Thus we have:

- Under the first definition (Def<sub>0</sub>), the quantity being calculated is simply the mean fitness distance. This idea carries over directly to the continuous case.
- Under the second definition (Def<sub>1</sub>), the idea is that mutations *should* create fitness distances of 1: lesser fitness distances are non-local, as are greater ones. In the continuous case we set up bounds,  $\alpha$  and  $\beta$ , and say that mutations *should* create fitness distances  $\alpha < fd < \beta$ . When  $fd < \alpha$ , we add  $\alpha - fd$  to  $d_m$ , penalising an overly-neutral mutation; when  $fd > \beta$ , we add  $fd - \beta$  to  $d_m$ , penalising a highly non-local mutation. Each of these conditions reflects a similar action in the discrete case.
- Under the third definition (Def<sub>2</sub>), both neutral and small non-zero fitness distances are regarded as local; only large fitness distances contribute to  $d_m$ . Thus, in the continuous case, only when  $fd > \beta$  do we add a quantity  $(fd - \beta)$  to  $d_m$ , penalising mutations of relatively large fitness distances.

From the previous new definitions on continuous-valued fitness, it is clear that it is necessary to know the “correct” values for  $\alpha$  and  $\beta$ . In Section 4 we discuss this in detail.

### 3 Related Work

Landscapes and problem difficulty have been the subject of a good deal of research in EC in general and GP in particular. Several approaches to investigating problem difficulty have been proposed. In this section we mention some of them, including their pros and cons.

#### 3.1 Sampling the Fitness Landscape

One of the best well-known prediction tools available in the literature is the one proposed by Jones called fitness distance correlation (*fdc*). This measures the



difficulty of a problem on the basis of the relationship between fitness and distance to the goal. The idea behind *fdc* was to consider fitness functions as heuristic functions and to interpret their results as indicators of the distance to the nearest optimum of the search space. *fdc* is an algebraic measure intended to express the degree to which the fitness function conveys information about distance to the optimum.

Altenberg [1] argued that predicting the hardness of a problem when using only fitness and distance in an EC system presents some difficulties. For instance, neither crossover nor mutation are taken into account when *fdc* is calculated, unless their effects are built-in to the measure of genetic distance used. Other works have also shown some weaknesses in *fdc*. Both [16] and [12] construct examples which demonstrate that the *fdc* can be “blinded” by particular qualities of the search space, and that it can be misleading. However, perhaps, the most important drawback of the *fdc* measure is that the global optimal solution must be known *a priori* in order to compute it, something that is basically impossible for real-world problems. There is, however, a vast amount of work where Jones’ approach has been successfully used. Of particular interest is the work by Vanneschi and colleagues [19], on the use of *fdc* in GP.

Later work by Vanneschi et al. attempted to address weaknesses of *fdc* with a new approach that is based on the concept of *fitness clouds* [18]. These are scatter plots that describe the manner in which the fitness of each individual correlates with the fitness of its neighbors. *Fitness clouds* are constructed by plotting a point for each genotype  $x$  on a 2-D plane, where the horizontal axis corresponds with the fitness of  $x$  given by  $f(x)$ , and the vertical axis represents the fitness  $f(y)$  of a neighboring genotype  $y$ . In this definition,  $y$  could be chosen in different ways, but the idea is to relate neighborhood with the search operators, particularly mutation. Given this scatter plot, Vanneschi et al. proposed the *negative slope coefficient (nsc)*, which is a measure that characterises the overall shape of the *fitness cloud* computed for a given problem. The *nsc* is computed by assuming a piecewise linear relationship between  $f(x)$  and  $f(y)$  for a sample of  $N$  individual genotypes and computing the slope of the scatter points within a set of equally spaced segments on the horizontal axis. The *nsc* is given in the range of  $(-\infty, 0]$ , where a value of 0 represents a very easy or evolvable problem, and a negative *nsc* indicates an increase in difficulty for the evolutionary search.

An important first observation is that *nsc* does not require the global optimum to be known *a priori*, a significant step foreword from *fdc*. However, *nsc* also exhibits some disadvantages. For instance, the scale of *nsc* is not clearly defined, in fact it is mostly described in informal terms. Moreover, while *nsc* is based on reasonable assumptions, the algorithm used to compute it relies on a series of *ad-hoc* design choices. Nonetheless, *nsc* is currently considered as one of the best predictor of search performance for GP.

**Table 1** Functions used as benchmark problems

$$\begin{aligned}
 &x + x^2 + x^3 \\
 &x + x^2 + x^3 + x^4 \\
 &x + x^2 + x^3 + x^4 + x^5 \\
 &x + x^2 + x^3 + x^4 + x^5 + x^6 \\
 &\sin(x^2)\cos(x) - 1
 \end{aligned}$$

## 4 Experimental Setup

### 4.1 Benchmark Problems

For our analysis, we have used five Real-Valued Symbolic Regression problems. Table 1 shows the functions used as benchmark problems in our study.

In simple terms, we can say that the goal of these type of problems is to find a program whose output is equal to the values of the specified functions. Thus, the fitness of an individual in the population must reflect how closely the output of an individual comes from the target. To compute the fitness of an individual it is common to define the fitness as the sum of absolute errors measured at different values of the independent variable (e.g.,  $x$ ), in this case in the range  $[-1.0, 1.0]$ . In this study we have measured the errors for  $x \in \{1.0, 0.9, 0.8, \dots, 0.8, 0.9, 1.0\}$  and we have defined an arbitrary threshold of 0.01. The latter is used to indicate that an individual with a fitness  $\leq$  threshold is regarded as a correct solution.

The problems are posed as maximisation tasks for an easier interpretation.

To study locality we need to have an alternative representation with differing locality and (presumably) differing performance. Therefore, we will use contrasts in encoding (two encodings: standard GP and a slightly modified encoding) and in operators (one-point and subtree mutation). The idea here is that each encoding will give a different value for locality and different performance, allowing us to compare the predictions of relative performance made by locality with results of evolutionary runs. This different encoding is presented next.

### 4.2 Uniform Genetic Programming

As mentioned previously, we expect to find different performance when using different encodings. This will allow us to compare the performance predictions made by locality with actual performance. So, we decided to adopt a slightly different tree-like structure representation, called Uniform GP (UGP) [7].

UGP (called uniform because all internal nodes are of the same arity) is an encoding defined by adding ‘dummy’ arguments (i.e., terminals/subtrees) to internal nodes whose arity is lower than the maximum arity defined in the function set. These are called dummy arguments because they are not executed when the individual is evaluated. For this to happen, it is necessary to use special functions that indicate the

use of dummy arguments (these special functions can be seen as flags that indicate that the subtree beneath these type of functions will not be executed). See [7] for details.

To use this representation in our five symbolic regression problems, we defined the function set:  $UGP = \{+, -, *, \%, Sin2, Cos2\}$ . Thus,  $Sin2, Cos2$  allow the addition of dummy arguments as explained previously. When using the typical GP representation we used the function set:  $GP = \{+, -, *, \%, Sin, Cos\}$ . For both  $UGP$  and  $GP$ ,  $\%$  is protected division.

### 4.3 Setting Bounds

In Section 2 we generalised the definition of GP locality in continuous-valued fitness. From  $Def_1$  and  $Def_2$  it is clear that it is necessary to establish the values needed for  $\alpha$  and  $\beta$ . In this work, we have used different values for these variables to determine the impact of these bounds on locality prediction.

This, in consequence, lead us to the use a several values for  $\alpha$  and  $\beta$ . So, in this work, we used four *starting* values for each variable ( $\alpha, \beta$ ) (i.e.,  $\alpha = \{0.01, 0.001, 0.0001, 0.00001\}$ ,  $\beta = \{0.1, 0.01, 0.001, 0.0001\}$ ). For each starting pair (i.e,  $Pair_{11} = \{\alpha = 0.01, \beta = 0.1\}$ ,  $Pair_{21} = \{\alpha = 0.001, \beta = 0.01\}$ ,  $Pair_{31} = \{\alpha = 0.0001, \beta = 0.001\}$ ,  $Pair_{41} = \{\alpha = 0.00001, \beta = 0.00001\}$ ), we automatically created 10 different values for each pair using different interval values (i.e., 0.01, 0.001, 0.0001, 0.00001). To create these 10 cases for each pair, we used one of the latter interval values for each of the starting pair values, and accumulatively, we added it to  $\alpha$  and  $\beta$  until all the cases were created. So, in total we defined 40 different configuration values for  $\alpha$  and  $\beta$  (e.g.,  $Pair_{12} = \{\alpha = 0.02, \beta = 0.11\}$ ,  $Pair_{22} = \{\alpha = 0.002, \beta = 0.011\}$ ,  $Pair_{32} = \{\alpha = 0.0002, \beta = 0.0011\}$ ,  $Pair_{42} = \{\alpha = 0.00002, \beta = 0.00011\}$ ).

### 4.4 Evolutionary Runs

The experiments were conducted using a GP with tournament selection (size = 7). To obtain more meaningful results, we performed 50 independent runs for each of the three configurations of population size and number of generations. Runs were stopped when the maximum number of generations was reached. Three population sizes were used (i.e., 200, 250, 500) along with three different number of generations (i.e., 125, 100, 50). Tournament selection (size 7) was used and we initialised our population using the ramped half-and-half method (depth 2 to 4). To control bloat we defined 1250 nodes and/or a maximum depth of 7. As indicated before, we used two different mutation operators: one-point and subtree. Finally, to obtain meaningful results, we performed 50 independent runs.

## 4.5 *Sampling and Measuring Locality*

To predict the performance of an encoding and an operator (as mentioned previously, we used one-point and subtree mutation), it is important to consider the type of sampling used.

As discussed previously, in his original studies, Rothlauf used bitstrings, fixed-length chromosomes to study the locality of a representation. So, in his work it was straightforward to create a sample to study the locality of a representation. This, however, changes when using a variable length, variable shape representation like in tree-like structures. So, one should pay special attention to this.

In our work, we have decided to create a sample based on actual evolutionary runs. That is, for each of the five problems, we saved all the individuals in the population in every generation along with their fitness values. Then, we sorted the individuals based on their fitness, and from there we sample uniformly from there. We sample 40,000 individuals for each of the problems used in this paper (i.e.,  $40,000 \cdot 5$ ). It is worth pointing out that we tried other sampling methods achieving similar results.

Now, to study and examine the locality present in each encoding (typical GP and UGP), for each data point in the sample data, we created an offspring via each mutation operator and measured locality using the three definitions introduced in Section 2. Thus, to compare the predictions made by locality (i.e., the lower the value on locality, the better the performance should be), we performed runs using the parameters described before in this section.

## 5 Results and Discussion

As mentioned in Section 2, Rothlauf [14] distinguished two forms of locality: high and non-locality (originally called “low-locality”). In a nutshell, the author claimed that a representation with high locality (the closer to 0, the higher the locality of the representation is, see the three definition introduced in Section 2) is more likely to perform an effective evolutionary search compared to a representation with non-locality (i.e., higher quantitative value). In the following paragraphs we discuss our findings in terms of how good or bad locality is as a prediction performance tool on continuous-valued problems.

### 5.1 *Quantitative Locality Measures and Performance*

To estimate how good or bad locality is in predicting performance in continuous fitness-valued cases, it is necessary to calculate quantitative locality measures using the three definitions introduced in Section 2 denoted by  $Def_0$ ,  $Def_1$  and  $Def_2$ . We know that  $Def_1$  and  $Def_2$  rely on two thresholds,  $\alpha$  and  $\beta$ . Now, one question arises:

**Table 2** Average of Quantitative Locality Measures on the Symbolic Regression Problems, using two encodings  $GP = \{+, -, *, \%, Sin, Cos\}$ ,  $UGP = \{+, -, *, \%, Sin2, Cos2\}$ , two mutations, and three locality definitions. *Lower is better.*

Mutation Operators	Def <sub>0</sub>		Def <sub>1</sub>		Def <sub>2</sub>	
	GP	UGP	GP	UGP	GP	UGP
$F_1 = x + x^2 + x^3$						
One Point	.0049	.0041	.0082	.0075	.0001	5.28e-05
Subtree	.0061	.0046	.0079	.0091	.0002	4.85e-05
$F_2 = x + x^2 + x^3 + x^4$						
One Point	.0042	.0045	.0080	.0074	.0004	.0008
Subtree	.0058	.0052	.0074	.0088	.0001	4.92e-05
$F_3 = x + x^2 + x^3 + x^4 + x^5$						
One Point	.0038	.0040	.0078	.0072	.0003	0.0006
Subtree	.0057	.0048	.0074	.0084	.0001	4.98e-05
$F_4 = x + x^2 + x^3 + x^4 + x^5 + x^6$						
One Point	.0056	.0052	.0079	.0074	.0002	.0005
Subtree	.0065	.0058	.0080	.0088	.0001	3.88e-05
$F_5 = \sin(x^2)\cos(x) - 1$						
One Point	.0044	.0035	.0079	.0071	.0003	.0008
Subtree	.0060	.0062	.0077	.0082	.0001	4.73e-05

what are the best values to use for these two thresholds? This is important to determine to know how sensitive each definition is. Thus, to address this question we tested different values for these two variables. That is, as mentioned in Section 4, we used different values for each of these variables and tested 40 different configurations values for  $\alpha$  and  $\beta$ . Due to space limitations and for clarity purposes 2, we decided to average all the quantitative measures for definitions Def<sub>1</sub> and Def<sub>2</sub>, for each of the five symbolic regression functions used in this work. These results are shown in Table 2. Recall that a low quantitative measure on locality denotes high-locality, whereas the opposite is true for non-locality.

To see how locality performed in predicting performance, it is necessary to perform evolutionary runs. Thus, we also performed extensive empirical experimentation (50 \* 30 \* 2 runs in total) 3, for each of the five regression problems used in this work. The performance (measured in terms of average of the best fitness values over all runs) are shown in Table 3. Now, it is necessary to examine if locality is able to correctly predict performance. To do so, we examine both locality and performance values, shown in Tables 2 and Table 3, respectively.

<sup>2</sup> We gathered extensive quantitative locality measures ( 40 \* 5 \* 3 \* 40,000 ) (10 different values for both  $\alpha$  and  $\beta$  using 4 different range of values for them, hence 40 different settings (see Section 4 for full details), 5 problems, 3 definitions of locality, and finally, we sample 40,000 individuals).

<sup>3</sup> 50 independent runs, 30 different settings (three different combinations of population sizes and number of generations, five different problems, and two different function sets for each of the five problems = 3 \* 5 \* 2), and two different mutation operators.

**Table 3** Performance (measured in terms of average of the best fitness values over all runs) of a Mutation-Based GP (using only one-point and subtree mutation) on the Symbolic Regression Problems. Numbers within parentheses indicate number of runs able to find the global optimum.  $GP = \{+, -, *, /, Sin, Cos\}$  and  $UGP = \{+, -, *, /, Sin2, Cos2\}$ . Higher is better.

Mutation	$Pop = 200,$ $Gen = 125$		$Pop = 250,$ $Gen = 100$		$Pop = 500,$ $Gen = 50$	
Operators	GP	UGP	GP	UGP	GP	UGP
	$F_1 = x + x^2 + x^3$					
One Point	.2417	.4831	.3769	.3619	.2788	.5966
			(3)	(4)		(17)
Subtree	.5540	.5229	.6496	.6420	.7308	.8958
	(10)	(19)	(21)	(18)	(29)	(41)
	$F_2 = x + x^2 + x^3 + x^4$					
One Point	.2276	.2788	.2180	.3161	.2318	.3208
Subtree	.9172	.4466	.4463	.6454	.4861	.6157
	(42)	(8)	(5)	(20)	(6)	(17)
	$F_3 = x + x^2 + x^3 + x^4 + x^5$					
One Point	.1574	.2768	.1730	.2205	.2031	.2316
Subtree	.3332	.3394	.3474	.3264	.4291	.3289
						(12)
	$F_4 = x + x^2 + x^3 + x^4 + x^5 + x^6$					
One Point	.1412	.1862	.1586	.1912	.1940	.1872
Subtree	.3161	.3089	.4339	.3005	.3780	.2846
			(1)			
	$F_5 = \sin(x^2)\cos(x) - 1$					
One Point	.4748	.5054	.4432	.5072	.4794	.6232
Subtree	.7329	.8716	.6020	.6289	.7514	.7995
	(1)	(1)	(1)	(1)		

So, let us start analysing the prediction done by locality  $Def_0$  (second column of Table 2) on the first function used ( $F_1 = x + x^2 + x^3$ ). We first focus our attention using One Point mutation. We take the best locality value (high locality is represented by a low quantitative measure as explained in Section 2). We can see that for this particular scenario, the best locality is achieved by UGP (.0041), then we proceed to examine the performance, shown in Table 3, achieved when using One Point mutation on  $F_1$  for the three different configurations of population ( $Pop = \{200, 250, 500\}$ ) and generations ( $gen = \{125, 100, 50\}$ ). So, according to locality  $Def_0$ , the best performance should be achieved by UGP. When we examine the performance achieved on the function  $F_1$  using one point mutation, we can see that in two out of three cases (i.e.,  $Pop = 200, Gen = 150, Pop = 500, Gen = 50$ ), locality  $Def_0$  correctly predicted performance. Now we proceed to do the same examining subtree mutation. For this scenario ( $Def_0$ ), we can see that the best locality values is achieved again by UGP (.0046). So, according to this, UGP should achieve better performance in the three different settings of population size and generations. When we check this in Table 3, we can see that this is the case only for one case ( $Pop = 500, Gen = 50$ ). Thus, summarising this, we have that for  $F_1$ ,  $Def_0$  of locality

**Table 4** Number of correct predictions of *good locality values* on performance (measured as mean best fitness over 100 runs), for the five Symbolic Regression problems and using two different mutation operators. The three different definitions of locality are denoted by Def<sub>0</sub>, Def<sub>1</sub> and Def<sub>2</sub>. Table 2 shows the locality definitions for continuous-valued fitness. Table 3 shows the performance achieved by GP and UGP in each of the five problems used in this work.

	One Point	Subtree	Total
$F_1 = x + x^2 + x^3$			
Def <sub>0</sub>	2	1	3
Def <sub>1</sub>	2	2	<b>4</b>
Def <sub>2</sub>	2	1	3
$F_2 = x + x^2 + x^3 + x^4$			
Def <sub>0</sub>	0	2	2
Def <sub>1</sub>	3	1	<b>4</b>
Def <sub>2</sub>	0	2	2
$F_3 = x + x^2 + x^3 + x^4 + x^5$			
Def <sub>0</sub>	0	1	1
Def <sub>1</sub>	3	2	<b>5</b>
Def <sub>2</sub>	0	1	1
$F_4 = x + x^2 + x^3 + x^4 + x^5 + x^6$			
Def <sub>0</sub>	2	0	2
Def <sub>1</sub>	2	3	<b>5</b>
Def <sub>2</sub>	1	0	1
$F_5 = \sin(x^2)\cos(x) - 1$			
Def <sub>0</sub>	3	0	3
Def <sub>1</sub>	3	0	3
Def <sub>2</sub>	0	3	3

was able to correctly predict three out of six scenarios (i.e., three different settings for population size and generations, for each of the mutation operators).

To help the reader to interpret the prediction done only by the “best” locality (i.e., lowest quantitative locality values), we process, as described in the previous paragraph, the data shown in Tables 2 and 3. Thus, in Table 4 we show the number of correct predictions done by each of the definitions of locality using one point and subtree mutation. From this, it is clear that best definition of locality able to correctly predict performance more frequently compared to the other definitions, is Def<sub>1</sub> (highlighted in boldface). It is, however, fair to say that it is not perfect. That is, the highest number of times able to correctly predict performance is five out of six cases (three settings for population size and generations, for each of the two mutation operators), which were the cases for  $F_3$  and  $F_4$ .

Of course, there are different ways to compare the prediction done by locality and the performance achieved by performing evolutionary search. For instance, one can consider *all* locality values, instead of only considering the best locality values, as we did before in Table 4. Also, instead of taking into consideration the mutation operators, now we compare locality values (see Table 2) and encoding used

**Table 5** Number of correct predictions of all locality values on performance (measured as mean best fitness over 100 runs), for the five Symbolic Regression problems and using two different mutation operators. The three different definitions of locality are denoted by Def<sub>0</sub>, Def<sub>1</sub> and Def<sub>2</sub>. Table 2 shows the locality definitions for continuous-valued fitness. Table 3 shows the performance achieved by GP and UGP in each of the five problems used in this work.

	GP	UGP	Total
	$F_1 = x + x^2 + x^3$		
Def <sub>0</sub>	3	3	6
Def <sub>1</sub>	4	4	<b>8</b>
Def <sub>2</sub>	3	3	6
	$F_2 = x + x^2 + x^3 + x^4$		
Def <sub>0</sub>	2	2	4
Def <sub>1</sub>	4	4	<b>8</b>
Def <sub>2</sub>	2	2	4
	$F_3 = x + x^2 + x^3 + x^4 + x^5$		
Def <sub>0</sub>	1	1	2
Def <sub>1</sub>	5	5	<b>10</b>
Def <sub>2</sub>	1	1	2
	$F_4 = x + x^2 + x^3 + x^4 + x^5 + x^6$		
Def <sub>0</sub>	2	2	4
Def <sub>1</sub>	5	5	<b>10</b>
Def <sub>2</sub>	1	1	2
	$F_5 = \sin(x^2)\cos(x) - 1$		
Def <sub>0</sub>	3	0	3
Def <sub>1</sub>	3	0	3
Def <sub>2</sub>	0	3	3

(GP and UGP). Let us start our analysis with Def<sub>0</sub> and  $F_1$ . We know that when we consider GP, the locality value is .0049 which is the “worst” locality (compared to .0041 achieved by UGP – recall that the closer to 0, the higher the locality is). This indicates, that GP should come in second place, regardless of the mutation operator used. When we compare this against actual performance (see Table 3), we can see that this happens in three out of six cases (three settings for population size and generations, and two mutation operators). Now, when we turn our attention to the same scenario (Def<sub>0</sub> and  $F_1$ ), but considering UGP, we can see that it has a better locality (.0041). This means that the performance should be better compared to GP (.0049). Again, when we compare this against performance (Table 3), we can see that locality was able to correctly predicts only three out of six cases. Table 5 summarises this analysis. We know that a perfect prediction for this scenario is 12<sup>4</sup> for each definition of locality. The summary shown in Table 5 indicates that the best locality definition is Def<sub>1</sub>, which agrees with our previous finding (see Table 4).

<sup>4</sup> 12 = 2 encodings \* 3 different setting for population size and generations \* 2 locality values (one for each encoding).



## 5.2 Definitions of Locality and Limitations

From the previous analysis, it is clear that locality can successfully be used as an estimation of problem difficulty. In particular,  $\text{Def}_1$  appears to be more accurate in predicting performance, as shown in Tables 4 and 5, but why does this happen? What are the key differences between the three definitions of locality (introduced in Section 2)? To answer these questions, one really needs to carefully analyse each definitions of locality. That is, it seems that the best results are obtained when a finer grained analysis is taken. For instance,  $\text{Def}_0$  only considers a rough estimate of average fitness divergence. While  $\text{Def}_2$  only considers large divergence using a single threshold ( $\beta$ ). On the other hand,  $\text{Def}_1$ , adjusts for “highly neutral mutations” and non-local changes, a more detailed analysis that yields the best results.

Thus, the results presented here suggest that locality is a more or less good indicator of expected performance, for continuous fitness-valued cases. However, there are still some shortcomings and open questions that should be addressed in future research. Particularly, it is worth mentioning that to calculate quantitative locality measures, a sampling method is necessary, as mentioned in Section 4. This is similar to *nsc* and *fdc*, where these measures rely on an extensive sampling procedure, which can be computationally costly. It appears that, for practical applications, the locality results should be evaluated based on their generalisation to a whole class of problems and not just single problem instances.

## 6 Conclusions

In this work, we made an effort to shed some light on how locality in Genetic Programming can predict performance. In particular, in continuous-fitness valued cases. For this purpose, we extended the original definition of locality proposed by Rothlauf [14] when he used bitstring from the genotype-phenotype mapping to the genotype-fitness mapping. We also, studied three different locality definitions, to determine if one definition of locality should be preferred among others.

To see if locality was able correctly predict performance, we use two different representations (i.e., typical GP, and a slightly different representation called Uniform GP), based on the assumption that each representation will give different performance.

Thus, is it possible to use locality as a prediction performance tool? This question has been discussed before (e.g., [4, 5, 14]). We argue that it is possible to get a good estimate on the performance of evolutionary runs by using locality. The key lessons we have learnt during our experiments are the following. Firstly, we have shown how it is possible to extend the definition of locality from a bitstring representation to a more complex representation (GP) for continuous fitness-valued cases. Secondly, we studied three different definitions of locality and proved that the definitions that require the use of threshold values (i.e.,  $\text{Def}_1$  and  $\text{Def}_2$ ) are robust

to their parametrisation). Thirdly, one definition of locality ( $\text{Def}_1$ ) achieved a higher accuracy compared to the other two definitions explored in this paper. All of these findings are based on an extensive amount of experimental evidence.

## References

1. Altenberg, L.: Fitness Distance Correlation Analysis: An Instructive Counterexample. In: Back, T. (ed.) *Proceedings of the Seventh International Conference on Genetic Algorithms*, San Francisco, CA, USA, pp. 57–64 (1997)
2. Beyer, H., Schwefel, H.: *Evolution strategies - A comprehensive introduction*. *Natural Computing* 1(1), 3–52 (2002)
3. D’haeseleer, P., Bluming, J.: Effects of locality in individual and population evolution. In: Kinnear, K.E. (ed.) *Advances in Genetic Programming*, pp. 177–198 (1994)
4. Galván-López, E., McDermott, J., O’Neill, M., Brabazon, A.: Defining locality in genetic programming to predict performance. In: *2010 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8 (2010)
5. Galván-López, E., McDermott, J., O’Neill, M., Brabazon, A.: Towards an understanding of locality in genetic programming. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010*, New York, NY, USA, pp. 901–908 (2010)
6. Galván-López, E., McDermott, J., O’Neill, M., Brabazon, A.: Defining locality as a problem difficulty measure in genetic programming. *Genetic Programming and Evolvable Machines* 12(4), 365–401 (2011)
7. Galván-López, E., Poli, R.: An empirical investigation of how degree neutrality affects GP search. In: Aguirre, A.H., Borja, R.M., García, C.A.R. (eds.) *MICAI 2009*. LNCS, vol. 5845, pp. 728–739. Springer, Heidelberg (2009)
8. Galván-López, E., Poli, R., Kattan, A., O’Neill, M., Brabazon, A.: Neutrality in evolutionary algorithms... what do we know? *Evolving Systems* 2(3), 145–163 (2011)
9. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading (1989)
10. Jones, T.: *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque (1995)
11. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge (1992)
12. Naudts, B., Kallel, L.: A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 4(1), 1–15 (2000)
13. Poli, R., Galván-López, E.: The effects of constant and bit-wise neutrality on problem hardness, fitness distance correlation and phenotypic mutation rates. *IEEE Trans. Evolutionary Computation* 16(2), 279–300 (2012)
14. Rothlauf, F.: *Representations for Genetic and Evolutionary Algorithms*, 2nd edn. Physica-Verlag (2006)
15. Stadler, P.F., Stephens, C.R.: Landscapes and Effective Fitness. *Comments on Theoretical Biology* 8(4), 389–431 (2003)
16. Tomassini, M., Vanneschi, L., Collard, P., Clergue, M.: A study of fitness distance correlation as a difficulty measure in genetic programming. *Evolutionary Computation* 13(2), 213–239 (2005)

17. Trujillo, L., Martínez, Y., Galván-López, E., Legrand, P.: Predicting problem difficulty for genetic programming applied to data classification. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011, New York, NY, USA, pp. 1355–1362 (2011)
18. Vanneschi, L., Clergue, M., Collard, P., Tomassini, M., Vérel, S.: Fitness clouds and problem hardness in genetic programming. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 690–701. Springer, Heidelberg (2004)
19. Vanneschi, L., Tomassini, M., Collard, P., Clergue, M.: Fitness Distance Correlation in Structural Mutation Genetic Programming. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 455–464. Springer, Heidelberg (2003)
20. Wright, S.: The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution. In: Jones, D.F. (ed.) Proceedings of the Sixth International Congress on Genetics, vol. 1, pp. 356–366 (1932)

# Analysis and Classification of Epilepsy Stages with Genetic Programming

Arturo Sotelo, Enrique Guijarro, Leonardo Trujillo\*,  
Luis Coria, and Yuliana Martínez

**Abstract.** Epilepsy is a widespread disorder that affects many individuals worldwide. For this reason much work has been done to develop computational systems that can facilitate the analysis and interpretation of the signals generated by a patient's brain during the onset of an epileptic seizure. Currently, this is done by human experts since computational methods cannot achieve a similar level of performance. This paper presents a Genetic Programming (GP) based approach to analyze brain activity captured with Electrocorticogram (ECoG). The goal is to evolve classifiers that can detect the three main stages of an epileptic seizure. Experimental results show good performance by the GP-classifiers, evaluated based on sensitivity, specificity, prevalence and likelihood ratio. The results are unique within this domain, and could become a useful tool in the development of future treatment methods.

**Keywords:** Epilepsy Diagnosis, Genetic Programming, Classification.

## 1 Introduction

Epilepsy is a neurological disorder that causes chronic seizures as part of its symptomatology. Some estimates state that the number of people that suffer from

---

Arturo Sotelo

Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico de Tijuana, Blvd. Industrial y Av. ITR Tijuana S/N, Mesa Otay C.P. 22500, Tijuana B.C., México  
e-mail: [soteloo@yahoo.com](mailto:soteloo@yahoo.com)

Enrique Guijarro

Departamento de Ingeniería Electrónica, Universidad Politécnica de Valencia, Spain  
e-mail: [eguijarro@eln.upv.es](mailto:eguijarro@eln.upv.es)

Leonardo Trujillo · Luis Coria · Yuliana Martínez

Doctorado en Ciencias de la Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico de Tijuana, Blvd. Industrial y Av. ITR Tijuana S/N, Mesa Otay C.P. 22500, Tijuana B.C., México  
e-mail: [leonardo.trujillo.ttl,luis.coria,ysaraimr}@gmail.com](mailto:{leonardo.trujillo.ttl,luis.coria,ysaraimr}@gmail.com)

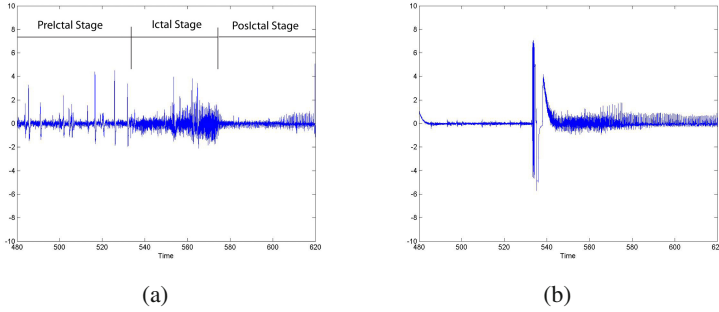
\* Corresponding author.

this disorder ranges between 11/100,000 to 134/100,000 [6], or that 1% to 5% of the general population experiences one or more seizures during their life-time [3, 20]. From the group of people with this disorder, two thirds can be treated by anti-epileptic medication and 7 or 8% can be cured by surgery [17]. Unfortunately, however, the symptomatology of the rest cannot be controlled by currently available therapies.

An epileptic seizure is a sudden episode that disrupts mental functions, motor control, sensorial abilities and autonomic activity. This is caused by a paroxysmal malfunction of brain cells, which is considered an abnormal increase of neural synchrony [15]. Epilepsy can affect a patient's brain partially or completely, respectively inducing partial or generalized seizures [22]. A seizure develops over several basic stages [11], these are: (1) the Basal stage (2) the Pre-Ictal Stage, (3) the Ictal stage; and (4) the Post-Ictal stage. The Basal stage represents normal brain activity, the waveform of brain signals during this stage are characterized by a low amplitude and a relative high frequency. In the Pre-Ictal stage, an Electroencephalography (EEG) or Electroocorticogram (ECoG) can show considerable amplitude increase relative to the Basal stage, with spikes and transitory activity, but no distinguishable symptoms can be seen in a patient during this stage. The Ictal stage is when the seizure occurs, producing jerky movements, olfactory sensations and even the loss of consciousness, depending if it is focal or generalized, brain signals are distinguished by high amplitude discharges, a low frequency and a predominant rhythm. The last stage is called Pos-Ictal, where signal recordings show general amplitude depression and a gradual return towards the Basal stage when symptoms cease.

If an expert neurologist analyzes the EEG or ECoG signal of a patient undergoing an epileptic seizure he can identify the seizure stages as they occur over time. For instance, Figure 1(a) depicts an ECoG signal taken over an entire episode, where the three main stages of the seizure are clearly marked. From this example it is clear that each stage is characterized by a different signal morphology. While a human expert has no problem identifying each stage, to our knowledge an automatic method for stage detection has not been developed. Nonetheless, other works have focused on predicting the onset of a seizure by identifying specific signal features [1, 13, 25]. An important aspect of most works in this area is that they focus on a small number of test subjects. Primarily because different patients tend to exhibit different signal patterns, even if they all share a similar general structure [12]. Therefore, while each stage is identifiable when you analyze the time-series of a seizure as a whole, if only a single two-second segment is considered, for example, then determining the stage to which it belongs is not trivially done.

In this work, we present an approach that can automatically determine the stage to which a signal segment belongs. The problem is posed as a supervised learning task, where the system takes as input a signal sample of a certain duration and from this determines the corresponding epileptic stage. However, deriving automatic processing methods for these signals is definitely not a straightforward endeavor, given the complexities of the brain signals generated during a seizure [4, 12]. In this paper the task is solved using a Genetic Programming (GP) classifier, that analyzes



**Fig. 1** ECoG signal of a level-5 seizure on Racine scale, showing how signal amplitude varies through time. **(a)** Deep recording through the stimulus electrode. **(b)** Seizure recorded by the cortex electrode.

basic statistical features of each signal and derives a non-linear mapping following a symbolic regression approach. Classifiers, similar to the ones derived here, could be used as computational tools that can assist a human expert during the analysis or diagnosis of epileptic signals. An even more ambitious goal could be to use these classifiers as part of an implanted device, that can monitor and react, in real-time, when a patient is experiencing the onset of specific stages of an epileptic seizure. However, such technological implementations are left as future lines of research.

The remainder of this paper proceeds as follows. Section 2 presents a brief introduction to Electrocorticogram signals from epileptic seizures. Then, Section 3 describes the ECoG dataset used in this study. Afterwards, Section 4 gives a formal description of the learning problem posed in this work and of the GP approach proposed to solve it. Section 5 presents the experimental setup and provides a detailed discussion of the main results. Finally, a summary of the paper and concluding comments are outlined in Section 6.

## 2 Epilepsy Signals

Normally, epileptic seizures occur spontaneously, a significant limitation to properly studying them. Therefore, in research work seizures are induced in a controlled experiment that use rodent test subjects, referred to as models. One of the most common is the amygdale kindling model, a model for temporal lobe epilepsy, the most common in human adults [19]. This model is used in the present work, since it is possible to induce self sustained seizures in rodents when required. This allows for the study of alternative treatments for drug-resistant human partial complex and secondarily generalized seizures [19].

In general, Electroencephalography (EEG) is the main tool for analysis and diagnosis of many neurological disorders. Brain activity produces a highly non-periodical signal with amplitude in the range of  $0.5\mu V - 100\mu V$ . Such signals can be detected by non-invasive methods when they are recorded at scalp level using EEG. However, EEG signals are normally contaminated by undesirable noise or artifacts produced by unrelated muscle or organ activity, which increases the difficulty of correctly interpreting such signals using automatic computational methods.

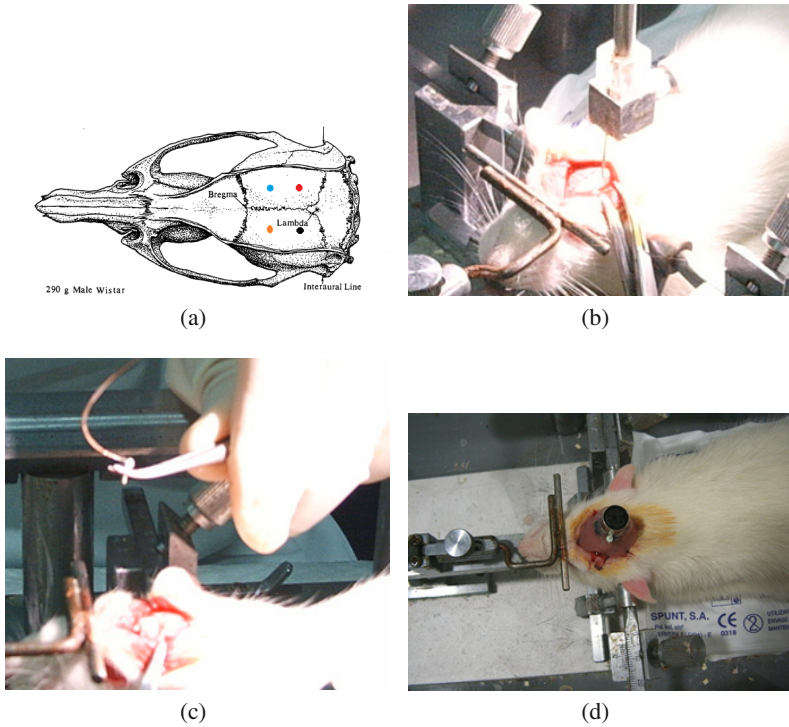
Less distorted signals can be obtained using intracranial recording methods through an Electrocorticogram (ECoG), which are far more amendable to precise analysis and clinical evaluations [8]. Intracranial detection can be accomplished by inserting needles within the brain at the required depths, these are called deep electrodes. Of course, the main drawback of such methods is the fact that they require surgical access to a patient's brain, a strong limitation for human test subjects. However, an advantage is that electrodes are bidirectional, and can be used for both detection and direct stimulation of the brain. In the former case, brain activity during a seizure can be intracranially recorded, free of artifacts [28], using metallic electrodes inserted within the cortex (an ECoG signal). In the latter, an electrode can be used to stimulate the brain and, if done correctly, to induce a seizure, as will be described in the following section.

As stated above, an epileptic seizure exhibits various stages as it develops; here we focus on the Pre-Ictal, Ictal and Post-Ictal stages. This paper presents an approach to automatically discriminate between these three stages within an epileptic signal, based on the local dynamics and morphology of a recorded ECoG signal from elicited epileptic episodes.

### 3 Experimental Data

It is normally unfeasible to record an epileptic seizure, since it is difficult to predict the onset of a seizure. Therefore, for research purposes artificially induced seizures provide valuable experimental data. Using animal models, it is possible to simulate chronic brain dysfunction that leads to epilepsy, a strategy that has allowed for research regarding the underlying causes and mechanisms behind epilepsy [9]. In particular, animal models are a valuable tool to study temporal lobe epilepsy [7, 19].

The Kindling model is used to study epilepsy that is induced by electrical impulses delivered to a previously healthy (non epileptic) animal. Epileptic conditions are achieved in the animal as the result of applying short duration electrical stimulus in the limbic regions of the brain, such as the amygdale or hippocampus. The amygdale kindling model in rats is considered the most appropriated for the study of alternative epilepsy treatments for partial and generalized seizures [2]. Through the Kindling model, spontaneous seizures are elicited by an electrical stimulus discharged directly to the brain of the rodent. The approach has several advantages, such as: first, precise focal activation; and second, a chronic epileptogenesis is reliably developed [22]. Kindling seizures are rated, depending on their symptoms,



**Fig. 2** (a) Illustration of approximate stereotaxic locations of stimulus, recording, and reference electrodes in a adult male Wistar rat [23]. (b) Implantation of the stimulus electrode through the rat's skull using a stereotaxic fixture. (c) Implantation the cortical recording electrode. (d) Final connector assembled on top of the rat's skull.

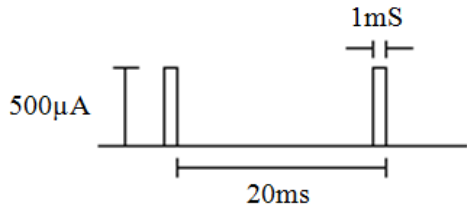
into a five level scale, know as the Racine scale [24]. This scale rates seizure intensity, from focal to generalized, depending on the symptomatology exhibited by rats from the Wistar breed, where level-5 is the highest intensity. Symptom for the five levels are: (0) No seizure response; (1) Immobility, eye closure, twitching of vibrissae; (2) Head nodding associated with more severe facial clonus; (3) Clonus of one forelimb; (4) Bilateral forelimb clonus with rearing; and (5) Rearing and falling on the back accompanied by generalize clonic seizures. In the present study, level-5 seizures (generalized motor seizures) are used for the experimental analysis.

### 3.1 Signal Recording

The electrode implants, the kindling experiments using live rodents (Wistar rats), and signal recording were carried out at the Centro de Investigación, Hospital General Universitario de Valencia, in Valencia, Spain. Stimulation and signal recording were achieved by inserting electrodes within the rodent's skull through symmetric



**Fig. 3** The stimulus signal applied to elicit an epileptic seizure on the rodent subject



burr holes at stereotaxic locations in accordance with [2, 14, 18]. Figure 2(a) shows the approximated stereotaxic location of the electrodes, where the black marks represent stimulus electrodes, orange are for reference, and blue and red represent the cortical frontal and occipital recording electrodes.

Stimulation was achieved through an electrode made of twisted pair of Teflon-coated 0.25 mm diameter stainless steel wires separated by 0.5 mm at the tip and 8 mm in length, and implanted through a burr hole, as shows in Figure 2(b). Two stainless steel screws served as cortical recording electrodes, as shown in Figure 2(c), these were attached to a connector assembly, as shown in Figure 2(d). After this process was done, the stimulation of rodent subjects began after 7 days.

The electric stimulation of the subjects brain tissue and deep recording of the epilepsy signal can be done after the electrodes are implanted and the connector is plugged in. Stimulation and recording of brain activity begins as soon as the rodent is connected, so the rodent does not remove the cable. The applied stimulus consist of a  $500\mu\text{A}$  @50Hz rectangular signal with a 5% duty cycle by 1s; the signal is depicted in Figure 3.

Electrical manifestations are of variable amplitude, with useful frequency components from 0.5 Hz to 60 Hz [26], and may find useful components up to 100 Hz [21] or 400 Hz [5]. For this work, the signal was bandpass filtered in a 0.5 Hz to 100Hz bandwidth, sampling rate was 256 Hz to avoid aliasing, using 12 bit resolution. The duration of a completely recorded seizure sometimes can be as much as 3 minutes. In some cases, more than a single stimulus needs to be applied to induce a seizure with a level-5 rating, since here we discard any seizure below this rating. Figure 1(a) shows an ECoG for a level-5 seizure.

Figure 1(a) presents the complete time-series record for a seizure, from the Pre-Ictal stage that begins at second 480 and ends at second 535. Then, the Ictal stage continues up to second 575, and finally the Post-Ictal stage represents the final part of the signal. The plot of 1(b) is the signal from the deep recording electrode, that is used as a reference to determine when the seizure is about to start. When the seizure is detected at the cortex level, this means that stimulus has produce an afterdischarge capable of stimulating the nearby neurons up to the cortex, producing a generalize seizure in the rodent. However, in some experiments the deep recording shows epileptic activity, but the cortex does not, which represents a local or focalized seizure.

## 4 Problem Statement

In this paper, the goal is to detect the three main seizure stages described above (Pre-Ictal, Ictal and Post-Ictal) given a short segment of a ECoG signal recorded using a cortex electrode. This problem can be posed as a classification task, where the signal segment represents a pattern  $\mathbf{x} \in \mathbb{R}^n$ , where  $n$  is the total number of sample points given a particular signal duration. For instance, since the sampling rate during recording is 256 Hz, if we take a 2 second signal then  $n = 512$ . Then, it is possible to construct a supervised learning problem where a training set  $\mathcal{X}$  of  $n$ -dimensional patterns with a known classification are used to derive a mapping function  $g(\mathbf{x}) : \mathbb{R}^n \rightarrow M$ , where  $M$  are the three distinct classes, in this case the three epilepsy stages.

This work uses a single test subject, a single rodent on which the seizures are induced and the signals recorded. This is partially justified due to the intra-patient variability that is usually observed in epileptic seizures [12]. Nonetheless, future work will focus on deriving classifiers that generalize across multiple patients or possible groups of them.

For this test subject, call him subject  $S_A$ , a level-5 seizure is induced and recorded on five consecutive days, call them Day-1, Day-2, Day-3, Day-4 and Day-5. Afterwards, the signal is classified manually by a human expert, who specifies where each epilepsy stage begins and ends, this provides the ground-truth for the learning problem. The signal is divided into  $N$  number of segments, each constituting a sample from the corresponding stage. All signal segments have the same duration, here we build two different datasets, using segments of 1 second and 2 seconds respectively. When the signal is divided, we allow for a slight overlap between consecutive segments, given by 20% of the total duration. Finally, it is important to state that signal segments that lie on two adjacent stages are removed from the dataset.

Then, the problem we pose can be stated as follows. The goal is to use the signal samples, or segments, from a single day, and use them as the learning data for the classifier. Then, the classifier is tested on the samples from the remaining four days. Therefore, the question is: if the signal from a single seizure is given, can it be used to train a classifier that is able to correctly detect the different signal stages from seizures from the same subject that are recorded on different days?

### 4.1 Proposal

The above problem is solved using a Genetic Programming (GP) classification system. GP can be used in various ways to solve such supervised classification tasks, see for instance [10, 16]. However, the approach proposed by Zhang and Smart [29] is used here, referred to as the Probabilistic GP Classifier, or PGPC for short [27, 29]. In PGPC, it is assumed that the behavior of  $h$  can be modeled using multiple Gaussian distributions, each corresponding to a single class [29]. The distribution of each class  $\mathcal{N}(\mu, \sigma)$  is derived from the examples provided for it in set  $\mathcal{X}$ ,

**Table 1** Parameters for the PGPC system used in the experimental tests

Parameter	Description
Population size	200 individuals.
Generations	200 generations.
Initialization	Ramped Half-and-Half, with 6 levels of maximum depth.
Operator probabilities	Crossover $p_c = 0.8$ ; Mutation $p_\mu = 0.2$ .
Function set	$\{+, -, *, /, \sqrt{\cdot}, \sin, \cos, \log, x^y,  \cdot , \text{if}\}$
Terminal set	$\{x_1, \dots, x_i, \dots, x_P\}$ Where each $x_i$ is a dimension of the data patterns $\mathbf{x} \in \mathbb{R}^P$
Bloat control	Dynamic depth control.
Initial dynamic depth	6 levels.
Hard maximum depth	20 levels.
Selection	Lexicographic parsimony tournament
Survival	Keep best elitism

by computing the mean  $\mu$  and standard deviation  $\sigma$  of the outputs obtained from  $h$  on these patterns. Then, from the distribution  $\mathcal{N}$  of each class a fitness measure can be derived using Fisher's linear discriminant; for a two class problem it proceeds as follows. After the Gaussian distribution  $\mathcal{N}$  for each class is derived, a distance is required. In [29], Zhang and Smart propose a distance measure between both classes as

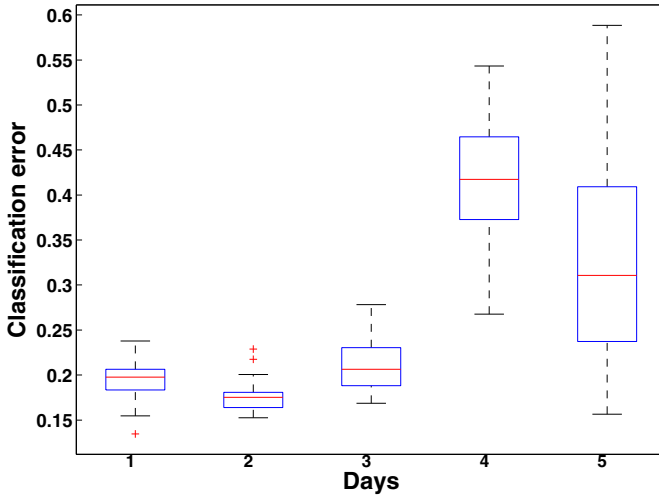
$$d = \frac{|\mu_1 - \mu_2|}{\sigma_1 + \sigma_2}, \quad (1)$$

where  $\mu_1$  and  $\mu_2$  are the means of the Gaussian distribution of each class, and  $\sigma_1$  and  $\sigma_2$  are their standard deviations. When this measure tends to 0, it is the worst case scenario because the mapping of both classes overlap completely, and when it tends to  $\infty$ , it represents the optimal case with maximum separation. To normalize the above measure, the fitness for an individual mapping  $h$  is given by

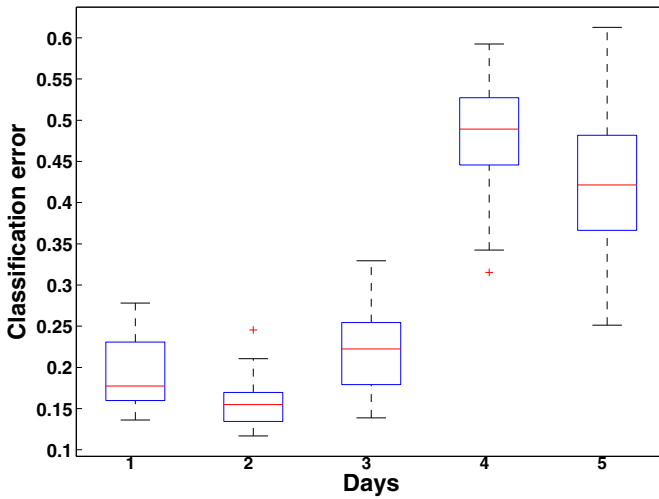
$$f_d = \frac{1}{1+d}. \quad (2)$$

After executing the GP, the best individual found determines the parameters for the Gaussian distribution  $\mathcal{N}_i$  associated to each class. Then, a new test pattern  $\mathbf{x}$  is assigned to class  $i$  when  $\mathcal{N}_i$  gives the maximum probability.

In summary, a GP classifier (PGPC) is trained using the epilepsy signal recorded during a single day, from a total of five different days, and then tested on the remaining days. The signal from each day is divided into segments of equal duration; two different partitions are built, the first has 1 second segments and the other uses segments with a duration of 2 seconds. The next section presents the experimental setup and main results.



(a) PGPC, 1s



(b) PGPC, 2s

**Fig. 4** Boxplots of the average classification error (y-axis), computed using different recording sessions for training (x-axis)

## 5 Experiments and Results

The PGPC algorithm uses a standard Koza-style GP, with a tree based representation, subtree-crossover and sub-tree mutation. The basic parameters are presented in Table 1. The terminal elements are basic statistical features computed for each signal segment  $x$  that is to be classified. Specifically, the terminal set  $T$  contains:

mean value  $x_\mu$ , median  $x_m$ , standard deviation  $x_\sigma$ , maximum  $x_{max}$ , minimum  $x_{min}$  and skewness  $x_s$ .

The total number of experiments are summed up as follows. Two different segment lengths (1 and 2 seconds) and five different signals used for training (5 different recording days), a total of (2x2x5) 20 different configurations. Moreover, 30 different runs are performed to obtain statistically significant results.

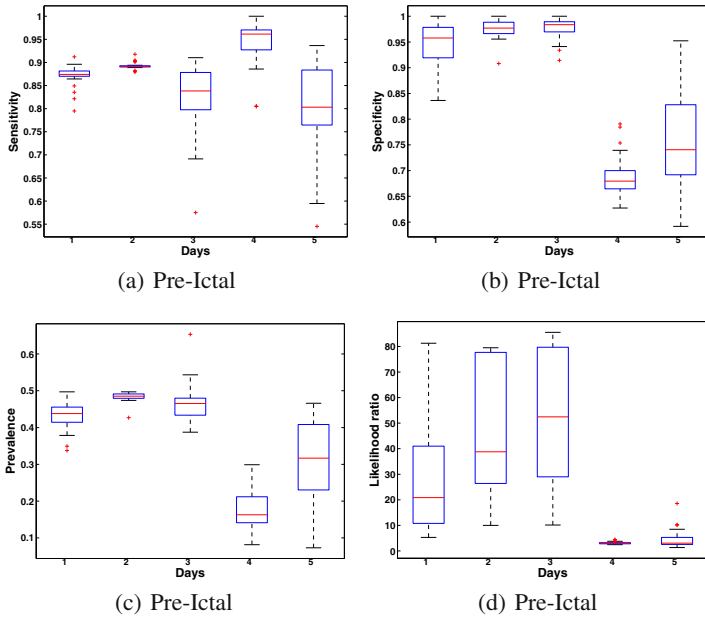
Figure 4 summarizes the results regarding the average test error for each configuration. The figure shows boxplots of the average classification error from each run, relative to the signal used for training (Day). The caption of plot states the GP algorithm used (PGPC) and the segment duration used to build the datasets (1 or 2 seconds). The algorithm is susceptible to the signal used for training, in general the signals from Day-4 and Day-5 produce the worst results. Also, signal length has a slight effect on classification error, with two second segments the classifier achieves better (less error) results. This difference does not seem to be substantial, but to maintain the following discussion compact we focus our analysis on classifier performance with 2 second segments.

To gain a deeper understanding of the performance achieved by the classifier, a detailed analysis of the results is presented using four standard performance indices: sensitivity, specificity, prevalence and likelihood ratio. However, given the results shown in Figure 4, only the results for the 2 second segments are analyzed. These measures are derived from the confusion matrix (true positives (TP), true negatives (TN), false positives (FP), false negatives (FN)) generated by the classifier with respect to each class; each index is computed as follows:

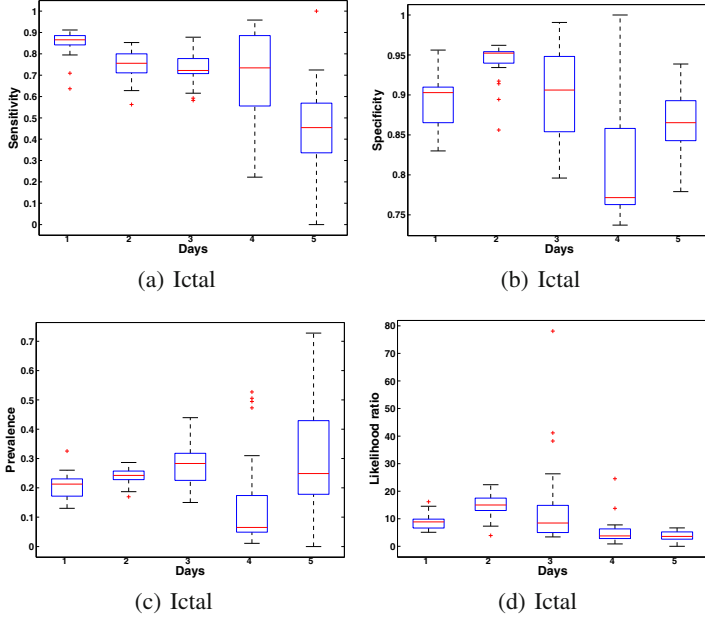
- Sensitivity:  $S = \frac{TP}{TP + FN}$ .
- Specificity:  $Sp = \frac{TN}{FN + TN}$ .
- Prevalence:  $P = \frac{TP + FP}{total}$ .
- Likelihood Ratio:  $LR_+ = \frac{S}{1 - Sp}$ .

Figures 5, 6 and 7 presents boxplots that summarize the results regarding the above performance indices computed for the PGPC classifier. Figures 5 corresponds to the values computed relative to the pre-ictal stage. Similarly, Figures 6 presents the values for the ictal stage, and Figures 7 corresponds to the post-ictal stage. Moreover, the following figures correspond to each index, sensitivity (a), specificity (b), prevalence (c) and likelihood ratio (d).

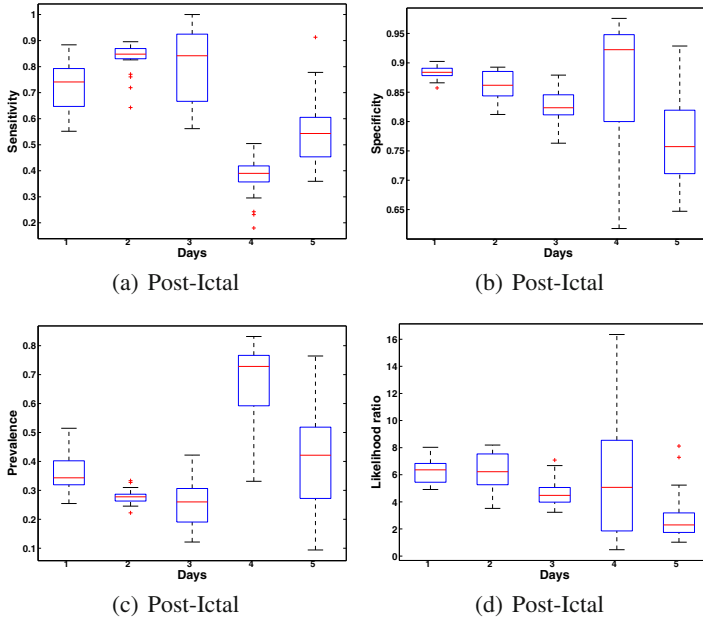
The results suggest that the GP classifier achieves good identification of epilepsy stages. These figures also show how performance depends on the signal used for training, with the best results achieved with the signals from Days 1 - 3, and the worst with Days 4 and 5. This result exhibits how seizures vary for individual subjects; nonetheless, the high performance achieved here is quite promising. This is confirmed by the high sensitivity and specificity achieved, with median values above 85% and 90% respectively for most configurations, a confident classification of



**Fig. 5** Boxplots that summarize the results for the PGPC classifier regarding Sensitivity (a), Specificity (b) , Prevalence (c) and Likelihood Ratio (d); with respect to the Pre-Ictal stage



**Fig. 6** Boxplots that summarize the results for the PGPC classifier regarding Sensitivity (a), Specificity (b) , Prevalence (c) and Likelihood Ratio (d); with respect to the Ictal stage



**Fig. 7** Boxplots that summarize the results for the PGPC classifier regarding Sensitivity (a), Specificity (b), Prevalence (c) and Likelihood Ratio (d); with respect to the Post-Ictal stage

random signal segments. Meanwhile, the low prevalence values, between 20% and 60%, shows that GP classifier can discard samples with a high confidence. Finally, the likelihood ratio reaches values larger than unity for all stages, the most promising result. The worst results were seen for the Post-Ictal stage, mostly attributable to the similarity it exhibits with the Pre-Ictal stage. If instead of classifying a random segment, the time series was classified progressively, then this shortcoming could be resolved. In general, the results show that the signal features extracted by the GP classifiers are highly discriminative and representative of each epilepsy stage.

## 6 Summary and Conclusions

This paper presents an approach that can automatically detect the corresponding epilepsy stage of random signal segments recorded by means of Electroencephalogram (EEG). This is done by posing a supervised learning problem, where an epileptic signal captured on a single day is used as training data, and the classifier is then tested on signal segments from five other recordings taken on different Days. The proposed approach is based on a GP-based classifier called the Probabilistic GP Classifier (PGPC) [29]. Experimental results are encouraging, based on the classification error, sensitivity, specificity, prevalence and likelihood ratio of the evolved classifiers. Moreover, since the classifiers are composed of basic mathematical

operations, given the terminal and functional primitives used, it is simple to implement them, in hardware or software, as part of an implanted device for real time monitoring or treatment. In general, these results are unique within the problem domain, and can become a useful tool in the development of future treatment technologies for epilepsy patients.

**Acknowledgements.** The authors thank the Departamento en Ingeniería Eléctrica y Electrónica from the Instituto Tecnológico de Tijuana.

## References

1. Daand, M., Esteller, R., Vachtsevanos, G., Hinson, A., Echaz, J., Litt, B.: Epileptic seizure prediction using hybrid feature selection over multiple intracranial eeg electrode contacts: A report of four patients. *IEEE Trans. Biomedical Engineering* 50(5), 603–615 (2003)
2. Barcia, J., Rubiuo, P.: Anticonvulsant and neurotoxic effects of intracerebroventricular injection of phenytoin, phenobarbital and arbamazepine in an amygdala-kindling model of epilepsy in the rat. *Epilepsy Research* 33, 159–167 (1999)
3. Barcia, J., Rubiuo, P.: Anticonvulsant and neurotoxic effects of intracerebroventricular injection of phenytoin, phenobarbital and carbamazepine in an amygdala-kindling model of epilepsy in the rat. *Epilepsy Research* 33, 159–539 (1999)
4. Bigan, C., Woolfson, W.: Time-frequency analysis of short segments of biomedical data. In: *IEEE Proceedings on Science, Measurement and Technology*, vol. 147(6), pp. 368–373 (2000)
5. Chiu, A., Jahromi, S., Khosravani, H., Carlen, P., Bardakjian, B.: The effects of high-frequency oscillations in hippocampal electrical activities on the classification of epileptiform events using artificial neural networks. *Journal of Neural Engineering* 3(1), 9–20 (2006)
6. Cockerell, O.: *Epilepsy, current concepts* (2003)
7. Coulter, D., McIntyre, D., Loscher, W.: Animal models of limbic epilepsies: What can they tell us? *Brain Pathol.* 2(12), 240–256 (2002)
8. D’Alessandro, M., Vachtsevanos, G., Esteller, R., Echaz, J., Litt, A.K.: Spectral entropy and neuronal involvement in patients with mesial temporal lobe epilepsy. In: *International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences* (2000)
9. Durand, D., Bikson, M.: Suppression and control of epileptiform activity by electrical stimulation: a review. *Proceedings of the IEEE* 89(7), 1065–1082 (2001)
10. Eggermont, J., Kok, J.N., Kusters, W.A.: Genetic Programming for Data Classification: Partitioning the Search Space. In: *Proceedings of the 2004 ACM Symposium on Applied Computing, SAC 2004*, pp. 1001–1005. ACM, New York (2004)
11. Franaszczuk, P., Bergey, G.: Time-frequency analysis using the matching pursuit algorithm applied to seizures originating from the mesial temporal lobe. *Electroenceph. Clin. Neurophysiol.* 106, 513–521 (1998)
12. Franaszczuk, P.J., Bergey, G.K.: Time-frequency analysis using the matching pursuit algorithm applied to seizures originating from the mesial temporal lobe. *Electroenceph. Clin. Neurophysiol.* 106(6), 513–521 (1998)



13. Iasemidis, L., Shiau, D., Sackellares, J., Pardalos, P., Prasad, A.: Dynamical resetting of the human brain at epileptic seizures: Application of nonlinear dynamics and global optimization techniques. *IEEE Trans. Biomedical Engineering* 51(3), 493–506 (2004)
14. Jeub, M., Beck, H., Sie, E., Ruschenschmidt, C., Speckmann, E., Ebert, U., Potschka, H., Freichel, C., Reissmüller, E., Loscher, W.: Effect of phenytoin on sodium and calcium currents in hippocampal ca1 neurons of phenytoin-resistant kindled rats. *Neuropharmacology* 42(1), 107–116 (2002)
15. Jouny, C., Franaszczuk, P., Bergey, G.: Characterization of epileptic seizure dynamics using gabor atom density. *Clinical Neurophysiology* 114(3), 426–437 (2003)
16. Koza, J.R.: Genetic programming II: automatic discovery of reusable programs. MIT Press, Cambridge (1994)
17. Litt, B., Echauz, J.: Prediction of epileptic seizures. *The Lancet Neurology* 1(1), 22–30 (2002)
18. Loscher, W., Reissmüller, E.: Anticonvulsant effect of fosphenytoin in amigdala-kindled rats: Comparison with phenytoin. *Epilepsy Research* 30, 69–76 (1998)
19. Loscher, W., Rundfeldt, C.: Kindling as a model of drug-resistant partial epilepsy: selection of phenytoin-resistant and non-resistant rats. *J. Pharmacol.* 258, 438–489 (1991)
20. Marchesi, B., Stelle, A., Lopes, H.: Detection of epileptic events using genetic programming. *IEE* 3, 1198–1201 (1997)
21. Mingui, S., Scheuer, M.: Time-frequency analysis of high-frequency activity at the start of epileptic seizures. *Proceedings IEEE/EMBS* 3, 1184–1187 (1997)
22. Morimoto, K., Fahnstock, M., Racine, R.: Kindling and status epilepticus models of epilepsy: rewiring the brain. *Progress in Neurobiology* 73, 1–60 (2004)
23. Paxinos, G., Watson, C.: *The Rat Brain in Stereotaxic Coordinates*, 4th edn. Academic Press, Sydney (1986)
24. Racine, R.: Modification of seizure activity by electrical stimulation. ii motor seizure. *Clinical Neurophysiology* 32(3), 281–294 (1972)
25. Sackellares, J.: Seizure prediction. *Epilepsy Currents* 8(3), 55–59 (2008)
26. Teplan, M.: Fundamentals of eeg measurement. *Measurement Science Review* 2(2), 1–11 (2002)
27. Trujillo, L., Martínez, Y., Galván-López, E., Legrand, P.: Predicting problem difficulty for genetic programming applied to data classification. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011*, pp. 1355–1362. ACM, New York (2011)
28. Zaveri, H.: Time frequency representation of electrocortigrams in temporal lobe epilepsy. *IEEE Transactions on Biomedical Engineering* 39, 502–509 (1992)
29. Zhang, M., Smart, W.: Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification. *Pattern Recogn. Lett.* 27, 1266–1274 (2006)

# Disparity Map Estimation by Combining Cost Volume Measures Using Genetic Programming

Enrique Naredo, Enrique Dunn, and Leonardo Trujillo\*

**Abstract.** Stereo vision is one of the most active research areas in modern computer vision. The objective is to recover 3-D depth information from a pair of 2-D images that capture the same scene. This paper addresses the problem of dense stereo correspondence, where the goal is to determine which image pixels in both images are projections of the same 3-D point from the observed scene. The proposal in this work is to build a non-linear operator that combines three well known methods to derive a correspondence measure that allows us to retrieve a better approximation of the ground truth disparity of stereo image pair. To achieve this, the problem is posed as a search and optimization task and solved with genetic programming (GP), an evolutionary paradigm for automatic program induction. Experimental results on well known benchmark problems show that the combined correspondence measure produced by GP outperforms each standard method, based on the mean error and the percentage of bad pixels. In conclusion, this paper shows that GP can be used to build composite correspondence algorithms that exhibit a strong performance on standard tests.

**Keywords:** Stereo Vision, Disparity Map, Genetic Programming.

---

Enrique Naredo · Leonardo Trujillo

Doctorado en Ciencias de la Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico de Tijuana, Blvd. Industrial y Av. ITR Tijuana S/N,

Mesa Otay C.P. 22500, Tijuana B.C., México

e-mail: [leonardo.trujillo.ttl@gmail.com](mailto:leonardo.trujillo.ttl@gmail.com), [enriquenaredo@gmail.com](mailto:enriquenaredo@gmail.com)

Enrique Dunn

University of North Carolina, at Chapel Hill Campus Box 3175, Sitterson Hall, Chapel Hill, NC 27599-3175, USA

e-mail: [dunn@cs.unc.edu](mailto:dunn@cs.unc.edu)

\* Corresponding author.

## 1 Introduction

One of the most complex task that the human body achieves is the perception of its surrounding environment by means of its senses. Vision is probably the most important sense to interpret the surrounding 3-D world. It allows us to perform basic tasks, such as the inference of shape and depth of objects by knowing the differences between their projections on both eyes. Also, from simple abilities we can carry out more complex computations, such as recognizing and interpreting the context of our natural environment. Not content with interpreting our personal reality, humans have attempted to endow man-made objects with this ability. Thus, this is the goal of computer vision research, the recovery of 3-D information and a description of natural scenes from 2-D digital images. However, while this task appears natural and simple to us, developing this ability for artificial entities has proven to be a challenging endeavor. Therefore, the development of unorthodox and creative solutions can provide promising new insights [3].

GP is a powerful search and optimization paradigm from the field of evolutionary computation used for automatic program induction [8, 9, 7]. GP can be used to solve two tasks simultaneously. First, it searches for the desired functionality required to solve a particular problem. Second, GP can also determine the structure of the solution, something that simple parameter optimization, such as that done by a genetic algorithm (GA), cannot achieve.

Moreover, GP is a very flexible technique, a characteristic that has allowed researchers to apply it in various fields and problem domains [8, 9, 7]. In computer vision, for instance, GP has been used for object recognition [6, 4, 5], image classification [10, 19], feature synthesis [11, 16], image segmentation [15, 18], feature detection [22, 21, 12] and local image description [13, 14]. GP has been successfully applied to computer vision problems since it is easy to define concrete performance criteria and the development or acquisition of data-sets is now trivially done. However, most problems in computer vision remain open and most successful proposals rely on state-of-the-art machine learning and computational intelligence techniques.

In this paper, we focus on the well-known problem of stereo vision, one of the main research topics in this field. Particularly, we address the problem of dense stereo correspondence, using a hybrid approach that combines several well-known methods using genetic programming (GP). The goal is to find a (non-linear) combination of standard correspondence algorithms by posing a search and optimization problem based on benchmark ground truth data.

The remainder of this paper proceeds as follows. The following section gives a brief introduction to the stereo vision problem, focusing on the the estimation of a dense disparity map from a stereo image pair. Afterwards, the problem solved in this work is clearly defined and the proposed solution is outlined. Then, the experimental setup and results are presented and discussed. Finally, a section with conclusions and future work is given.

## 2 Stereo Computer Vision

The perception of depth from a scene by both eyes is better known as *stereopsis*. Both eyes create two different images due to their different position on the head. Similarly, in *stereo computer vision*, a calibrated stereo system of two cameras separated a certain distance, like two eyes, take pictures of the same scene and the images obtained this way are better known as a *stereo image pair* [20].

One of the main problems in stereo computer vision consists on determining the *correspondence* between both images, since they are projections of the same scene. The goal is to match every element within the right image to the corresponding elements on left image (or vice-versa), this frames the basic *stereo matching* or *correspondence* problem.

Even though there is a wide range of choices for stereo correspondence algorithms, at the moment there is not a general solution. In particular, dense stereo matching consider rectified images, obtained from a calibrated stereo system, where the quality of the correspondence algorithm is directly related to that of the images.

Therefore, the goal of this work is to improve the quality of the raw disparity map estimation, by combining three well known measures, rather than using just a single one. This combination is performed using Genetic Programming (GP), where the resultant, possibly non-linear, combination will be better than the standard individual methods.

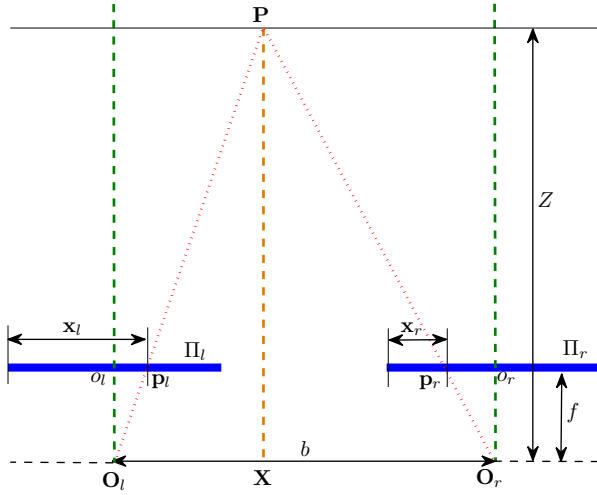
### 2.1 Disparity Map

The difference computed between a stereo pair of images taken from a calibrated camera system is called *disparity*. Let  $I_l$  and  $I_r$  respectively denote the left and right image of a stereo system. Then the difference between the position of corresponding pixels in both images is referred to as the *disparity* measured in pixel units. If, for instance,  $\mathbf{p}_r$  is a pixel in image  $I_r$ , and  $\mathbf{p}_l$  is an image pixel in image  $I_l$ , and  $\mathbf{p}_r$  and  $\mathbf{p}_l$  correspond to the same 3-D point in the imaged scene, then the disparity is given by;  $\mathbf{d} = |\mathbf{p}_r - \mathbf{p}_l|$ . This is depicted in Figure 1.

There is an inverse relation between disparity and depth, such a way objects that show more disparity between a stereo pair of images are closer to the observer. Therefore, disparity is seen as a three-dimensional projective transformation.

Since the camera arrangement only has a different horizontal position, there is a region  $R$  on the horizontal direction in the right image, where a search is performed to match an associated pixel from the left image, then the search region  $R(\mathbf{p}_l)$  depends on the left pixel  $\mathbf{p}_l$ . The model that describes the left image  $I_l(i, j)$  in terms of right image  $I_r(i, j)$  is

$$I_l(i, j) = I_r(i - d(1, j), j), \quad (1)$$



**Fig. 1** Canonical stereo system with focal lengths  $f$  and a baseline displacement  $b$ . The difference between coordinates  $x_l$  and  $x_r$  is called a (horizontal) disparity between points  $\mathbf{p}_l$  and  $\mathbf{p}_r$ .

where  $d(i, j) \geq 0$  is the displacement of each pixel; note that the image formed by these displacements is the disparity map  $d(i, j)$ .

For each correspondence method we can associate a cost function  $f$  that measures the similarity between pixel  $\mathbf{p}_l$  of image  $I_l$  and all of the pixels  $\mathbf{p}_r \in R$  with coordinates  $(i - d(1, j), j)$  from image  $I_r$ .

$$d(i, j) = \arg \min_{d'(i, j) \in R} \{f(i, j, d'(i, j))\}, \quad (2)$$

the measure  $f$  is generated by considering different values  $d \in [0, d_{max}]$ , with a cost volume  $CV$  such that, Equation 2 can be written as

$$d(i, j) = \arg \min_{d'(i, j) \in R} \{CV\}. \quad (3)$$

Because we focus in obtaining a raw disparity map, therefore, no filter method is applied to "clean up" spurious mismatches.

## 2.2 Correspondence Algorithms

Dense two frame correspondence algorithms are classified in two classes: correlation and feature-based methods [17]. Our research focuses on correlation-based

methods, where the elements to match are *image windows* of fixed size and the similarity criterion is a measure of the correlation between windows in both images. The corresponding element is given by the window that maximizes the similarity criterion within a search region.

Three well known correlation-based methods are Sum of Absolute Differences (SAD), Normalized Cross Correlation (NCC), and Birchfield-Tomasi (BTO), and they are used to compute the required cost volumes [2].

Sum of Absolute Differences (SAD), is a widely used correspondence algorithm for measuring the similarity between a pair of stereo images. It works by taking the absolute difference between each pixel in the left image  $I_l(i, j)$  and the corresponding pixel in the right image  $I_r(i, j)$ , These differences are summed to create a similarity measure.

For each pixel in the search region  $R$ , the cost volume  $CV_{SAD}$  is calculated for each sequence  $s$  of two stereo images, by computing the correlation value using;

$$SAD = \sum_{i=1}^M \sum_{j=1}^N |I_r(i, j) - I_l(i, j)|, \quad (4)$$

where the right image  $I_r$  of  $M$  by  $N$  pixels is correlated with a surrounding left image  $I_l$ , within the search region  $R$ , windows radius is of 3.

The Normalized Cross Correlation (NCC), uses the mean pixel value of both images  $I_l(i, j)$  and  $\bar{I}_l(i, j)$  to correlate pixels in the region search  $R$  from right image  $I_r$  with the left image  $I_l$ . For each pixel, the cost volume  $CV_{NCC}$  is calculated for each sequence  $s$  of two stereo images, by computing the correlation value using;

$$NCC = \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \frac{\{I_r(i, j) - \bar{I}_l(i, j)\} \{I_l(i, j) - \bar{I}_l(i, j)\}}{\sigma_{(l)} \sigma_{(r)}}, \quad (5)$$

where the right image  $I_r$  of  $M$  by  $N$  pixels is correlated with a surrounding left image  $I_l$ , within the search region  $R$ , windows radius is of 3.

Birchfield and Tomasi (BTO) [11], have proposed a matching cost that is insensitive to image sampling. Rather than just comparing pixel values shifted by integral amounts (which may miss a valid match), they compare each pixel in the reference image against a linearly interpolated function of the other image. For each pixel in the search region  $R$ , the cost volume  $CV_{BTO}$  is calculated for each sequence  $s$  of two stereo images, by computing the correlation value using;

$$BTO = N_{OCC} K_{OCC} - N_m K_r + \sum_i i = 1^N d(i, j), \quad (6)$$

where  $N_{OCC}$  is the number of occlusions (not the number of occluded pixels),  $K_{OCC}$  is a constant of occlusion penalization, technically interpreted as the evidence quantity necessary to declare a disparity change.  $N_m$  is the number of correspondences.  $K_r$  is the reward constant for every correspondence, is interpreted as the maximum

dissimilarity quantity between pixels, that generally is expected between two matching pixels,  $d(p_l, p_r)$  is the dissimilarity between pixels  $p_l$  and  $p_r$ .

### 3 Proposal

After the above introductory sections, the research goal of the present work is clearly defined in this section. Moreover, the proposed solution is presented and justified below.

#### 3.1 Problem Statement

In this work, the hypothesis is that three is better than one when it comes to correspondence algorithms. In other words, we wish to find an operator  $K$  that takes as input the cost volume produced by the three correspondence methods discussed above ( $CV_{SAD}$ ,  $CV_{NCC}$  and  $CV_{BTO}$ ) and generates as output a new cost volume  $CV_O$  which is optimal with respect to the accuracy of the respective disparity map. This task is posed as a supervised learning problem, where the ground truth disparity is known for a set of  $N$  images, the goal is to minimize the error of the disparity map given by  $K(CV_{SAD}, CV_{NCC}, CV_{BTO})$  with respect to the ground truth.

#### 3.2 Proposed Solution

This paper formulates the above problems in terms of search and optimization, with the goal of automatically generating an optimal operator  $K$ . The proposal is to solve this problem using a GP search, where the terminal (input) elements for each individual are the cost volumes  $CV_{SAD}$ ,  $CV_{NCC}$  and  $CV_{BTO}$ , and the output is  $CV_{GP}$ , the best possible approximation to  $CV_O$  found by the evolutionary process.

## 4 Experimental Configuration and Results

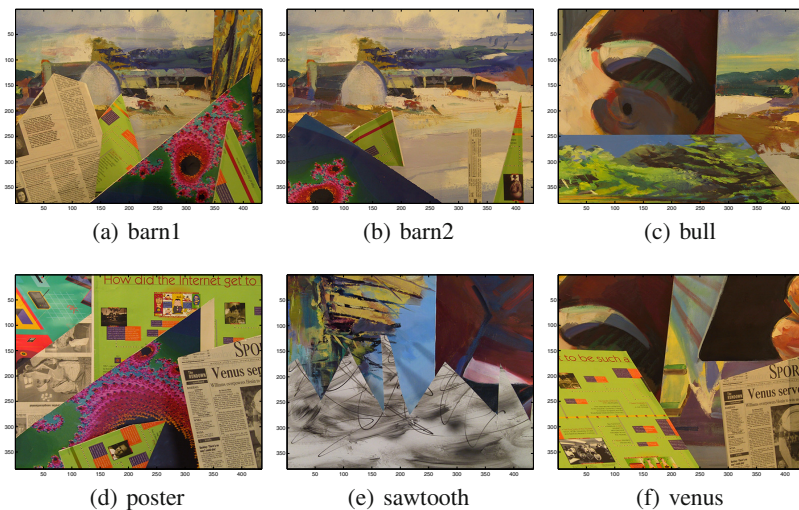
The proposal developed in this work is evolving an operator that combines three well known correspondence measures, using Genetic Programming.

### 4.1 Dataset

The images used on this work are taken from the Middlebury Stereo Vision Page [2](http://www.vision.middlebury.edu/stereo) which are standard benchmark tests for stereo vision algorithms. The dataset used contains six pairs of stereo images named: Barn1, Barn2, Bull, Poster, Sawtooth,

---

<sup>2</sup> [www.vision.middlebury.edu/stereo](http://www.vision.middlebury.edu/stereo)



**Fig. 2** Stereo images from [www.middlebury.edu/stereo](http://www.middlebury.edu/stereo) for experiment tests

and Venus. The left image from each pair is shown in Figure 2. Moreover, for each image pair the dataset includes both left-right and right-left pixel accurate disparity maps as ground truth.

## 4.2 GP Search

The goal of our experimental work is to assess the quality of the raw disparity map generated by GP, compared with respect to the base methods and the ground truth (GT) of each image pair.

In this work a standard Koza style tree-based GP representation is used, with subtree-crossover and subtree-mutation. The general parameters of the search are given in Table 4.2. Of particular importance is the terminal set, which is given by the cost volumes from the correspondence functions used: Sum of Absolute Differences (SAD), Normalized Cross Correlation (NCC) and Birchfield-Tomasi (BTO), therefore,  $CV_{SAD}$ ,  $CV_{NCC}$ ,  $CV_{BTO}$ , represent cost volumes from each method and are used as terminals for GP.

To compute each cost volume the search region depends on the minimum and maximum disparity given by the ground truth disparity map of the particular image; therefore, for Barn1 the search region is: [28,131], for Barn2 is: [27,132], for Bull: [29,153], for Poster: [27,161], Sawtooth has: [31,143], and Venus: [24,158]. Furthermore, the size of the neighborhood considered by each method is  $3 \times 3$ .



**Table 1** GP system parameters for the experimental tests

Parameter	Description
<i>Population size</i>	20 individuals.
<i>Generations</i>	100 generations.
<i>Initialization</i>	<i>Ramped Half-and-Half</i> , with 6 levels of maximum depth.
<i>Operator probabilities</i>	Crossover $p_c = 0.8$ ; Mutation $p_\mu = 0.2$ .
<i>Function set</i>	$\{+, -, \times, \div,  \cdot , x^2, \sqrt{x}, \log, \sin, \cos\}$ .
<i>Terminal set</i>	$\{CV_{SAD}, CV_{NCC}, CV_{BTO}\}$ . Where <i>CV</i> is the Cost Volume from SAD, NCC, and BTO functions respectively.
<i>Bloat control</i>	Dynamic depth control.
<i>Initial dynamic depth</i>	6 levels.
<i>Hard maximum depth</i>	20 levels.
<i>Selection</i>	Lexicographic parsimony tournament.
<i>Survival</i>	Keep best elitism.

Fitness is given by the cost function shown in Equation: [7](#) that assigns a cost value to every individual  $K$  expression as feasible solution given by GP. The goal is minimize the error computed between the disparity map from every  $K$  expression and the ground truth, therefore

$$f^S(K) = \frac{1}{NMS} \sum_{i=1}^M \sum_{j=1}^N |d_{(CV_m)}^S(i, j) - d_{(GT)}^S(i, j)|, \quad (7)$$

where  $s$  is the image sequence used,  $N$  is the total number of image pixels,  $(i, j)$  represents the pixel position on the matrix,  $d_{CV_m}^s$  is the disparity map computed using the method  $m$  (SAD, NCC, or BTO) for the  $s$  image sequence .

### 4.3 Experimental Setup

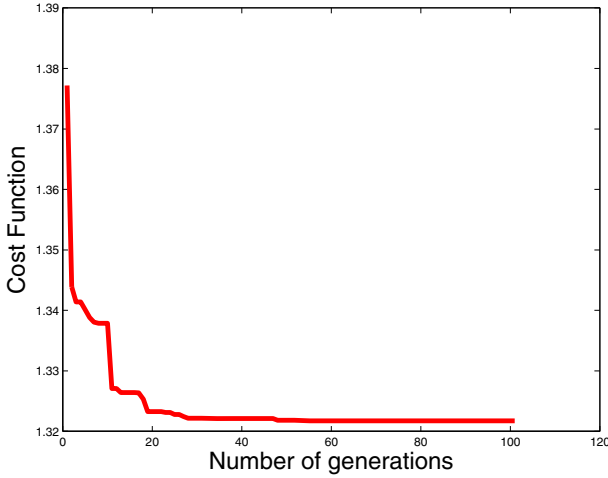
First three image sequences; Barn1, Barn2, and Bull, were selected as training data and the other three images sequences; Poster, Sawtooth, and Venus, as testing data, Besides the error evaluation explained in Eq. [7](#) we use to compare the error function showed on Middlebury University page that compute bad pixel percentage, given by

$$BP = \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N (|d_{CV_m}^S(i, j) - d_{GT}^S(i, j)| > \delta_d), \quad (8)$$

where the threshold for a bad evaluation  $\delta_d$  is a disparity error tolerance. For the experiments we use  $\delta_d = 1.0$ , since it is the threshold value used on the reference Middlebury web page.

#### 4.4 Experimental Results

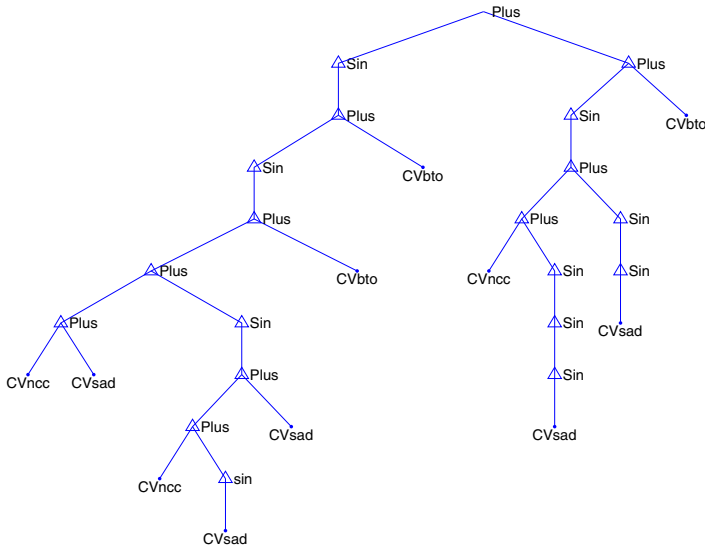
We compare the SAD, NCC, and BTO matching costs against the GP cost volume. The GP search was executed over 20 times, in all cases achieving similar performance. Figure 3 presents the convergence plot for the best run, which shows how the fitness of the best individual evolved over each generation. Figure 4 presents the best GP solution found expressed as a program tree. Moreover, the mathematical expression of the best GP individual  $K$  is given in Equation 9.



**Fig. 3** Convergence Graphic of the best GP run result

$$f_{gp}^* = \sin\{\sin[\sin(\sin CV_{SAD} + CV_{SAD} + CV_{NCC}) + CV_{SAD} + CV_{NCC} + CV_{BTO}] + CV_{BTO}\} + \sin(\sin^3 CV_{SAD} + \sin^2 CV_{SAD} + CV_{NCC}) + CV_{BTO}, \quad (9)$$

Finally, Tables 3-7 present a comparative analysis of the best GP individual and the three standard methods. Each table shows both the mean of the absolute difference, and the bad pixel percentage errors for each pair of stereo images. In all cases we can note that the GP method achieves the best performance. Figures 5-10 present a comparative visual analysis of the best GP individual and the three standard similarity measures along with the ground truth. These figures illustrate the improvement provided by the GP operator  $K$  over the standard methods.



**Fig. 4** Program tree of the best solution found by GP

**Table 2** Comparison results for the Barn1 image (bold indicates best result)

Error method	SAD	NCC	BTO	GP
Mean of Absolute Differences	1.483379	1.344098	1.834855	<b>1.311435</b>
Bad Pixel Percentage	1.021885	0.805163	1.397609	<b>0.771604</b>

**Table 3** Comparison results for the Barn2 image (bold indicates best result)

Error method	SAD	NCC	BTO	GP
Mean of Absolute Differences	1.959252	1.660405	2.038269	<b>1.640279</b>
Bad Pixel Percentage	1.579143	1.248889	1.774773	<b>1.226271</b>

**Table 4** Comparison results for the Bull image (bold indicates best result)

Error method	SAD	NCC	BTO	GP
Mean of Absolute Differences	1.785469	1.378297	2.030898	<b>1.330506</b>
Bad Pixel Percentage	1.186709	0.703539	1.422414	<b>0.675807</b>

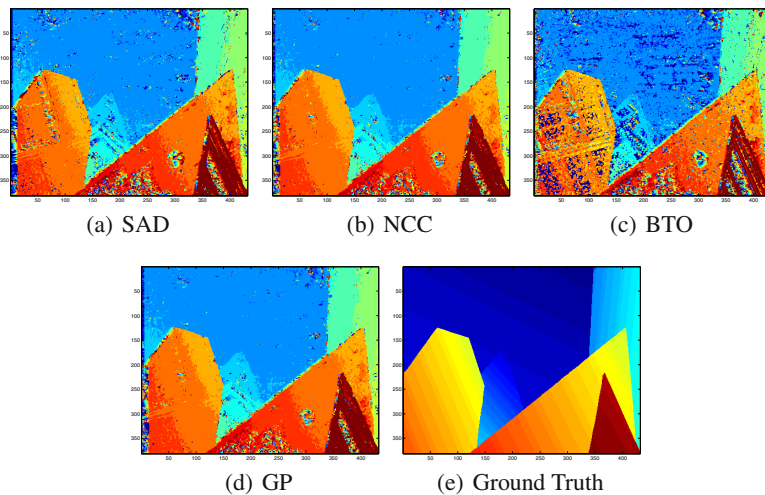


Fig. 5 Disparity maps comparison for the Barn1 image

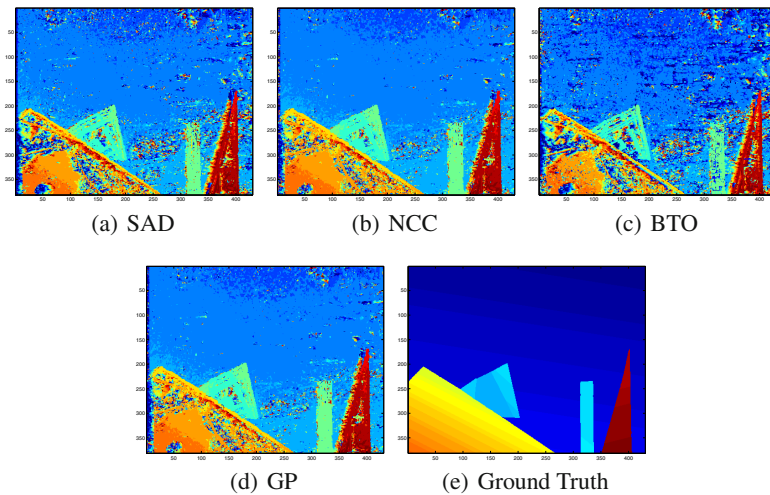
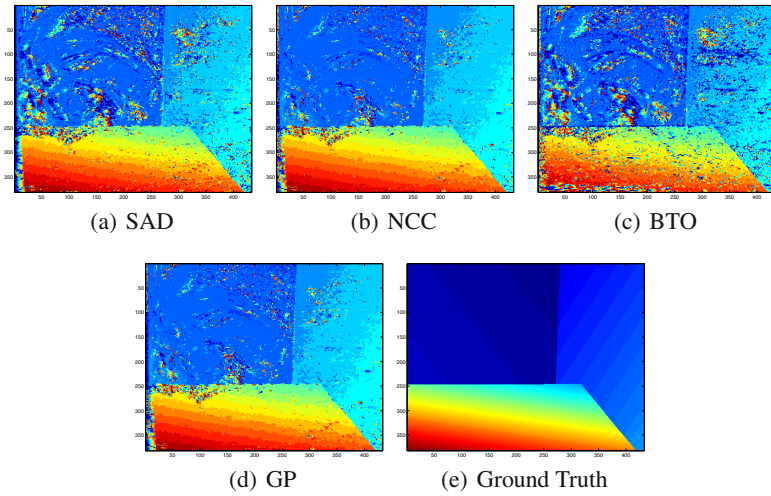


Fig. 6 Disparity maps comparison for the Barn2 image



**Fig. 7** Disparity maps comparison for the Bull image

**Table 5** Comparison results for the Poster image (bold indicates best result)

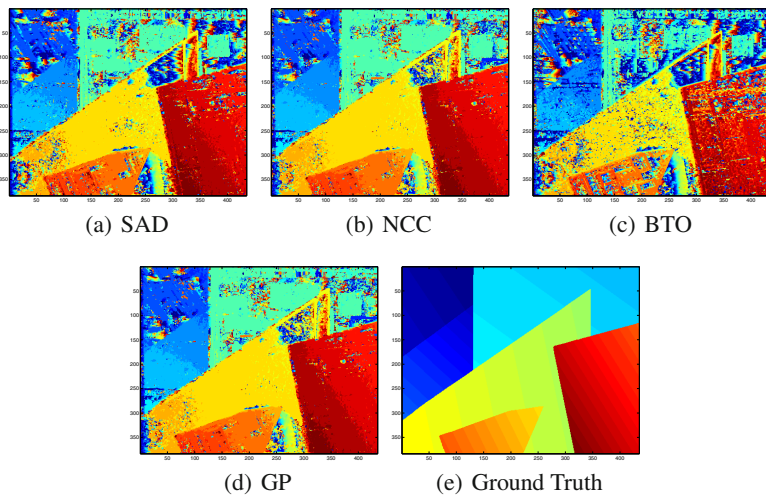
Error method	SAD	NCC	BTO	GP
Mean of Absolute Differences	2.432111	1.988050	3.010154	<b>1.945446</b>
Bad Pixel Percentage	1.57552	1.218279	1.985391	<b>1.202187</b>

**Table 6** Comparison results for the Sawtooth image (bold indicates best result)

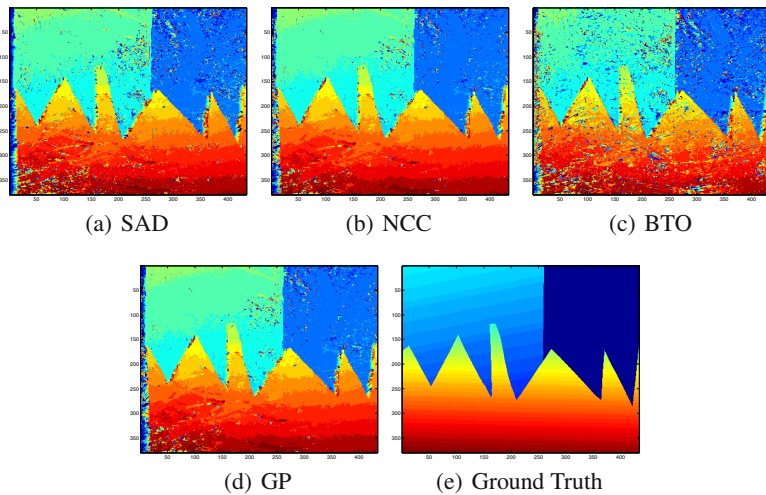
Error method	SAD	NCC	BTO	GP
Mean of Absolute Differences	1.532218	1.433264	1.727435	<b>1.374315</b>
Bad Pixel Percentage	1.073662	0.859555	1.169753	1.202187

**Table 7** Comparison results for the Venus image (bold indicates best result)

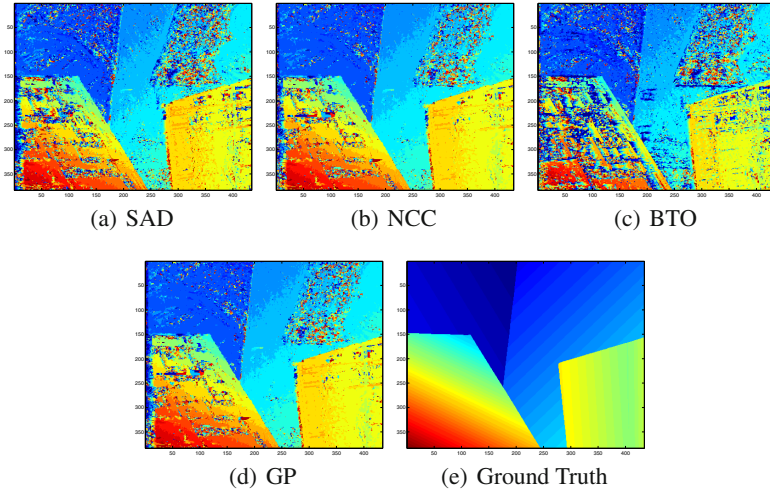
Error method	SAD	NCC	BTO	GP
Mean of Absolute Differences	2.281053	1.916733	2.704778	<b>1.880739</b>
Bad Pixel Percentage	1.617716	1.285286	1.804815	<b>1.267094</b>



**Fig. 8** Disparity maps comparison for the Poster image



**Fig. 9** Disparity maps comparison for the Sawtooth image



**Fig. 10** Disparity maps comparison for the Venus image

## 5 Conclusions and Future Work

This paper studies the problem of dense stereo correspondence using GP. The proposed approach is to combine three well-known similarity measures and derive a composed estimation of the disparity map for a stereo image pair. This task is posed a search and optimization problem and solved with GP. The terminal elements for the GP search were the SAD, NCC, BTO, and fitness is based on the error between the estimated disparity map and the ground truth disparity. Experimental results show that the evolved GP operator achieves better performance than the standard methods based on well-known benchmark problems. These results are validated with a set of test cases and an additional performance metric. While these results are an encouraging first step, further work is considering following this topic. For instance, we can add other similarity measures as input elements for the GP search, such as non-parametric correspondence methods. Moreover, we can use the raw disparity map generated by the GP operators as input data for global optimization methods which could allow us to define a higher level fitness evaluation.

**Acknowledgements.** The authors thank the Departamento en Ingeniería Eléctrica y Electrónica from the Instituto Tecnológico de Tijuana.

## References

1. Birchfield, S., Tomasi, C.: Depth discontinuities by pixel-to-pixel stereo. *Int. J. Comput. Vision* 35(3), 269–293 (1999)
2. Brown, M.Z., Burschka, D., Hager, G.D.: Advances in computational stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(8), 993–1008 (2003)
3. Cagnoni, S., Lutton, E., Olague, G.: Genetic and Evolutionary Computation for Image Processing and Analysis. In: EURASIP Book Series on Signal Processing and Communications, vol. 8. Hindawi Publishing Corporation (2008)
4. Ebner, M.: A real-time evolutionary object recognition system. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (eds.) EuroGP 2009. LNCS, vol. 5481, pp. 268–279. Springer, Heidelberg (2009)
5. Hernández, B., Olague, G., Hammoud, R., Trujillo, L., Romero, E.: Visual learning of texture descriptors for facial expression recognition in thermal imagery. *Computer Vision and Image Understanding, Special Issue on Vision Beyond the Visual Spectrum* 106(2-3), 258–269 (2007)
6. Howard, D., Roberts, S.C., Brankin, R.: Target detection in sar imagery by genetic programming. *Advances in Engineering Software* 30(5), 303–311 (1999)
7. Koza, J.: Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines* 11(3), 251–284 (2010)
8. Koza, J.R., Keane, M.A., Yu, J., Forrest, I., Bennett, H., Mydlowec, W.: Automatic creation of human-competitive programs and controllers by means of genetic programming. *Genetic Programming and Evolvable Machines* 1(1-2), 121–164 (2000)
9. Koza, J.R., Streeter, M.J., Keane, M.A.: Routine high-return human-competitive automated problem-solving by means of genetic programming. *Information Sciences* 178(23), 4434–4452 (2008)
10. Krawiec, K.: Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines* 3(4), 329–343 (2002)
11. Krawiec, K., Bhanu, B.: Visual learning by coevolutionary feature synthesis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 35(3), 409–425 (2005)
12. Olague, G., Trujillo, L.: Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming. *Image Vision Comput.* 29(7), 484–498 (2011)
13. Pérez, C.B., Olague, G.: Learning invariant region descriptor operators with genetic programming and the f-measure. In: 19th International Conference on Pattern Recognition (ICPR 2008), Tampa, Florida, USA, December 8-11, pp. 1–4. IEEE (2008)
14. Perez, C.B., Olague, G.: Evolutionary learning of local descriptor operators for object recognition. In: GECCO 2009: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 1051–1058. ACM, New York (2009)
15. Poli, R.: Genetic programming for feature detection and image segmentation. In: Forgarty, T.C. (ed.) AISB Workshop Evolutionary Computing, pp. 110–125 (1996)
16. Puente, C., Olague, G., Smith, S., Bullock, S., Hinojosa-Corona, A., González-Botello, M.: A genetic programming approach to estimate vegetation cover in the context of soil erosion assessment. *Photogrammetric Engineering and Remote Sensing* 77(4), 363–376 (2011)
17. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision* 47(1-3), 7–42 (2002)



18. Song, A., Ciesielski, V.: Texture segmentation by genetic programming. *Evol. Comput.* 16(4), 461–481 (2008)
19. Tan, X., Bhanu, B., Lin, Y.: Fingerprint classification based on learned features. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 35(3), 287–300 (2005)
20. Trucco, E., Verri, A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River (1998)
21. Trujillo, L., Legrand, P., Lévy-Véhel, J.: The estimation of hölderian regularity using genetic programming. In: *GECCO 2010: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 861–868. ACM, New York (2010)
22. Trujillo, L., Olague, G., Lutton, E., Fernández de Vega, F.: Multiobjective design of operators that detect points of interest in images. In: Cattolico, M. (ed.) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, Atlanta, GA, July 12-16, pp. 1299–1306. ACM, New York (2008)

**Part III**  
**Evolutionary Multi-objective Optimization**

# Finding Evenly Spaced Pareto Fronts for Three-Objective Optimization Problems

Heike Trautmann, Günter Rudolph, Christian Dominguez-Medina,  
and Oliver Schütze

**Abstract.** The averaged Hausdorff distance  $\Delta_p$  is a performance indicator in multi-objective evolutionary optimization which simultaneously takes into account proximity to the true Pareto front and uniform spread of solutions. Recently, the multi-objective evolutionary algorithm  $\Delta_p$ -EMOA was introduced which successfully generates evenly spaced Pareto front approximations for bi-objective problems by integrating an external archiving strategy into the SMS-EMOA based on  $\Delta_p$ . In this work a conceptual generalization of the  $\Delta_p$ -EMOA for higher objective space dimensions is presented and experimentally compared to state-of-the-art EMOA as well as specialized EMOA variants on three-dimensional optimization problems.

## 1 Introduction

In a variety of applications one is faced with the problem that several objectives have to be optimized concurrently leading to a *multi-objective optimization problem* (MOP). One important characteristic of such problems is that the solution set—the so-called Pareto set (PS)—typically forms a  $(d - 1)$ -dimensional object, where  $d \geq 2$  is the number of objectives of the MOP. Since it is rarely possible to determine

---

Heike Trautmann · Günter Rudolph

Fakultät Statistik, Technische Universität Dortmund, 44221 Dortmund, Germany

e-mail: [trautmann@statistik.tu-dortmund.de](mailto:trautmann@statistik.tu-dortmund.de),

[Guenter.Rudolph@tu-dortmund.de](mailto:Guenter.Rudolph@tu-dortmund.de)

Christian Dominguez-Medina

CINVESTAV-IPN, Depto. de Ingeniería Eléctrica, Sección de Computación, 07738 Mexico City, Mexico

e-mail: [hdomigueza@sagitario.cic.ipn.mx](mailto:hdomigueza@sagitario.cic.ipn.mx)

Oliver Schütze

Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, Col. San Pedro Zacatenco, 07360 Mexico City, Mexico

e-mail: [schuetze@cs.cinvestav.mx](mailto:schuetze@cs.cinvestav.mx)

the PS—or its image, the so-called Pareto front (PF)—analytically, it is desired to present the decision maker (DM) a ‘suitable’ finite size approximation of the set of interest, at least for low values of  $d \in \{2, 3\}$ . For this purpose, there already exist many methods for the numerical treatment of such problems. Among them, so-called evolutionary multi-objective algorithms (EMOAs) have caught the interest of many practitioners and researchers. Reasons for this fact include that these set oriented methods are applicable to a wide range of MOPs (e.g., regarding differentiability assumptions or other properties of the objectives such as multi-frontality or disconnected domains) and that they are capable of delivering a finite size approximation  $F$  of the Pareto front  $F^*$  in a single run of the algorithm. One important question is how the elements of  $F$  should be arranged in order to be a ‘suitable’ representation of  $F^*$  for the DM. This is certainly problem dependent, however, two desired properties that are mentioned most frequently are *convergence* and *spread*. That is, in many cases it is desired to obtain a set  $F$  such that all the entries are close to  $F^*$  (convergence) and are at the same time evenly distributed on  $F^*$  (spread). In the authors’ opinion, the most appropriate performance metric that combines these two goals simultaneously is the *averaged Hausdorff distance*  $\Delta_p$  [20]. As a consequence, in this work we are interested in an EMOA that aims for good (i.e., low)  $\Delta_p$  values of the outcome set  $F$  to the PF  $F^*$ . The main challenge in the design of such algorithms is apparently the efficient estimation of  $F^*$  since this set is not known a priori.

Recently, a first attempt has been made to design EMOAs that aim for averaged Hausdorff approximations of the PF [2]. The core of this work is a new external archiving strategy that aims for low  $\Delta_p$  values. In this strategy, the polygonal line obtained by the current set of ‘best’ (i.e., non-dominated) solutions together with the PL metric [3] is used in order to estimate the ideal Hausdorff approximation of the true PF. The latter is used to update the archive. The competitiveness of the resulting archiving approach, coupled with the SMS-EMOA [4] as generator, has been shown empirically on several benchmark problems. This method, however, is restricted to bi-objective problems (i.e.,  $d = 2$ ) due to the estimation of the PFs by polygonal lines.

In this work, we extend the EMOA presented in [2] to MOPs with more than two objectives, where we give special attention to the three-objective case. This generalization, however, is not straightforward, especially due to the required sequential generation of target fronts for the  $\Delta_p$  computation in the course of the generations based on the current PF approximation. In case of  $d \geq 3$  we therefore opt for mapping the  $d$ -dimensional objective vectors to 2-dimensional space via metric *Multi-Dimensional Scaling (MDS)* which eases the construction of the archiving strategy (in 2D). The individual with worst  $\Delta_p$  contribution to a set of grid points within the  $\alpha$ -convex hull of the population in MDS space is discarded within the archive update. In addition, the archive is now coupled with the EMOA by alternatingly selecting parents from the population and the archive in order to generate a wider

range of candidate points for the archive update as the SMS-EMOA tends to concentrate on knee regions and boundary points of the PF in the course of the algorithm run.

The remainder of this paper is organized as follows: In Sec. 2 we provide the required background for the understanding of this work. In Sec. 3 we describe the proposed methods which we compare to other state-of-the-art algorithms on four selected three-objective MOPs in Sec. 4. Finally, we draw our conclusions in Sec. 5 and outline paths for future work.

## 2 Background

In the following we consider unconstrained MOPs of the form  $\min\{f(x) : x \in \mathbb{R}^n\}$  where  $f(x) = (f_1(x), \dots, f_d(x))'$  is defined as the vector of the  $d \geq 2$  objective functions  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  for  $i = 1, \dots, d$ . The optimality of a MOP is defined by the concept of *dominance* [5].

### Definition 1

- Let  $v, w \in \mathbb{R}^d$ . If  $\forall i = 1, \dots, d : v_i \leq w_i$  then  $v \preceq w$  and  $v$  is termed less than or equal to  $w$ . If  $v \preceq w$  and  $v \neq w$  then  $v \prec w$  and  $v$  is said to be less than  $w$ .
- A vector  $y \in \mathbb{R}^n$  is said to be dominated by a vector  $x \in \mathbb{R}^n$  ( $x \prec y$ ) with respect to (MOP) if  $f(x) \prec f(y)$ , otherwise  $y$  is called non-dominated by  $x$ .
- A point  $x \in \mathbb{R}^n$  is called (Pareto) optimal or a Pareto point if there is no  $y \in \mathbb{R}^n$  that dominates  $x$ .
- The set of all Pareto optimal solutions  $X^*$  is called the Pareto set (PS).
- The image  $F^* = f(X^*)$  of  $X^*$  is termed the Pareto front (PF).

The set  $\text{ND}(X, \preceq) = \{x^* \in X \mid \nexists x \in X : x \prec x^*\}$  represents the *set of non-dominated elements relative to X*. Thus,  $F^* = \text{ND}(f(\mathbb{R}^n), \preceq)$ . Similarly, for some  $X \subseteq \mathbb{R}^n$  and  $f : X \rightarrow \mathbb{R}^d$  the set  $\text{ND}_f(X, \preceq) = \{x \in X : f(x) \in \text{ND}(f(X), \preceq)\}$  contains those elements from  $X$  whose images are nondominated in the image space  $f(X) = \{f(x) : x \in X\} \subseteq \mathbb{R}^d$ .

Since the PS and the PF can typically not be computed analytically, one task in multi-objective optimization is to numerically detect a finite size approximation of  $F^* = f(X^*)$ . So far, it is still an open question how an ‘optimal’ representation of the set of interest should look like for general MOPs. For this, several performance metrics have been proposed in order to measure the quality of the outcome set  $F$  of a search procedure such as an EMOA. We refer e.g. to [6, 7] for an overview of existing performance indicators. In this work we are particularly interested in a low distance between  $F$  and  $F^*$  and a sufficiently good spread. In particular, we are interested in the averaged Hausdorff distance  $\Delta_p$  as performance indicator which takes both criteria into account.

**Definition 2** Let  $u, v \in \mathbb{R}^n$ ,  $A, B \subset \mathbb{R}^n$ , and  $\|\cdot\|$  be a vector norm. The Hausdorff distance  $d_H(\cdot, \cdot)$  is defined as follows:

$$(a) \ d(u, A) := \inf_{v \in A} \|u - v\|$$

$$(b) d(B, A) := \sup_{u \in B} d(u, A)$$

$$(c) d_H(A, B) := \max(d(A, B), d(B, A))$$

The Hausdorff distance is widely used in many fields. It has, however, certain limitations when measuring the distance of the outcome of an EMOA to the PF since outliers generated by EMOAs are punished too strongly by  $d_H$ . As a remedy, we follow the suggestion of [20] and use the *averaged* Hausdorff distance.

**Definition 3** Let  $X = \{x_1, \dots, x_\mu\} \subset \mathbb{R}^n$  be a candidate set and  $Y = \{y_1, \dots, y_\mu\} \subset \mathbb{R}^d$  be its image, i.e.,  $y_i = f(x_i)$ ,  $i = 1, \dots, \mu$ . Further, let  $P = \{p_1, \dots, p_m\} \subset \mathbb{R}^d$  be a discretization of the PF. Then it is

$$\Delta_p(Y, P) = \max \left( \left( \frac{1}{\mu} \sum_{i=1}^{\mu} d(y_i, P)^p \right)^{1/p}, \left( \frac{1}{m} \sum_{i=1}^m d(p_i, Y)^p \right)^{1/p} \right)$$

The indicator  $\Delta_p$  can be viewed as a composition of slight variations of the Generational Distance (GD, see [23]) and the Inverted Generational Distance (IGD, see [5]). It is  $\Delta_\infty = d_H$ , but for finite values of  $p$  the indicator  $\Delta_p$  averages (using the  $p$ -vector norm) the distances considered in  $d_H$ . Hence, in spite of  $d_H$ ,  $\Delta_p$  does in particular not punish single (or few) outliers in a candidate set.

Another commonly accepted measure [10] for assessing the quality of an approximation is the so-called *dominated hypervolume* of a population.

**Definition 4** Let  $v^{(1)}, v^{(2)}, \dots, v^{(\mu)} \in \mathbb{R}^d$  be a nondominated set and  $r \in \mathbb{R}^d$  such that  $v^{(i)} \prec r$  for all  $i = 1, \dots, \mu$ . The value

$$H(v^{(1)}, \dots, v^{(\mu)}; r) = \Lambda_d \left( \bigcup_{i=1}^{\mu} [v^{(i)}, r] \right)$$

is termed the *dominated hypervolume with respect to reference point  $r$* , where  $\Lambda_d(\cdot)$  denotes the Lebesgue measure in  $\mathbb{R}^d$ .  $\square$

This measure has a number of appealing properties but determining its value is getting the more tedious the larger the number of objectives [11].

## 3 Methods

### 3.1 Base EMOA

The multi-objective evolutionary algorithms known as NSGA2 [12] and SMS-EMOA [4] use two selection methods sequentially: Firstly, the population is partitioned in a hierarchy of nondominated sets (termed *nondominated sorting*) which represents a ranking of individuals w.r.t. their degree of nondominance. Secondly,

the elements either with the least crowding distance (NSGA2) [12] or with the least hypervolume contribution (SMS-EMOA) [4] are discarded from the worst ranked subset that is needed to fill the population. The third base EMOA used for comparison is the MOEA/D [24] that works entirely different to the methods mentioned above. Since we modify only the SMS-EMOA in this paper a description of the base version and concepts is given below.

A population  $P$  can be partitioned in  $h$  disjunct nondominated sets  $R_1, \dots, R_h$  where  $h$  is the height of the partially ordered set  $P$ :

$$R_1 = \text{ND}_f(P, \preceq) \text{ and } R_k = \text{ND}_f\left(P \setminus \bigcup_{i=1}^{k-1} R_i, \preceq\right) \text{ for } k = 2, \dots, h \text{ if } h \geq 2.$$

This procedure is known by the term *nondominated sorting*. Evidently, every element from  $R_j$  is dominated by some individual in  $R_i$  if  $i < j$ .

The *hypervolume contribution* of some element  $x \in R_k$  is simply the difference  $H(R_k; r) - H(R_k \setminus \{x\}; r)$  between the dominated hypervolume of set  $R_k$  and the dominated hypervolume of set  $R_k$  without element  $x$ . Now we are in the position to describe the SMS-EMOA.

**Algorithm 1** Pseudo code of the SMS-EMOA

---

**SMS-EMOA =**

- 1: draw multiset  $P$  with  $\mu$  elements  $\in \mathbb{R}^n$  at random
  - 2: **repeat**
  - 3:   generate offspring  $x \in \mathbb{R}^n$  from  $P$  by variation
  - 4:    $P = P \cup \{x\}$
  - 5:   build ranking  $R_1, \dots, R_h$  from  $P$
  - 6:    $\forall i = 1, \dots, d : r_i = \max\{f_i(x) : x \in R_h\} + 1$
  - 7:    $\forall x \in R_h : h(x) = H(R_h; r) - H(R_h \setminus \{x\}; r)$
  - 8:    $x^* = \text{argmin}\{h(x) : x \in R_h\}$
  - 9:    $P = P \setminus \{x^*\}$
  - 10: **until** stopping criterion fulfilled
- 

## 3.2 Tools

The  **$k$  nearest neighbor (kNN)** concept is integrated into modified NSGA2 variants which are detailed in Sec. 3.3. The  $k$ th nearest neighbor  $y^{(k)}$  of a point  $y^{(0)} \in Y \subset \mathbb{R}^d$  is defined iteratively via

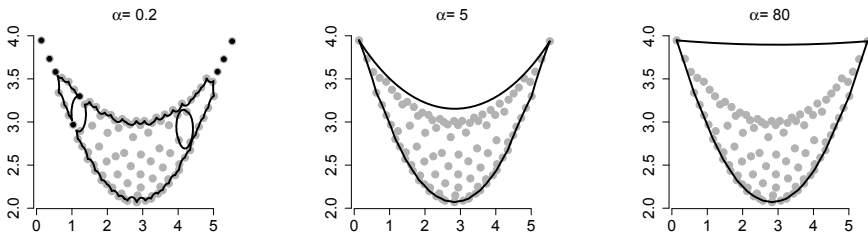
$$y^{(k)} = \text{argmin} \left\{ d(y, y^{(0)}) : y \in Y \setminus \bigcup_{j=0}^{k-1} \{y^{(j)}\} \right\} \text{ for } k = 1, 2, \dots, |Y| - 1$$

where  $d(\cdot, \cdot)$  is a metric on  $\mathbb{R}^d$ . Here, the Euclidean metric is used.

A method known as **metric multidimensional scaling (MDS)** [14] is integrated into the archiving mechanism of the  $\Delta_p$ -EMOA introduced in Sec. 3.3. MDS is a visualization technique which transforms a set of  $n$  points from a  $d$ -dimensional into a lower-dimensional euclidean space (usually 2D or 3D). It operates only on a matrix of given distances (e.g. euclidean) or dissimilarities  $d(i, j)$  between the points. The mapping is performed in such a way that the distances  $d(i, j)$  in the original space are maintained as closely as possible by solving the optimization problem

$$\min_{z_1, \dots, z_n} \sum_{i \neq j} (||z_i - z_j|| - d(i, j))^2 .$$

Here, the  $z_i \in \mathbb{R}^k$  with usually  $k = 2$  are the low-dimensional mappings of the original points  $x_i$ .



**Fig. 1** Example:  $\alpha$ -convex hull for increasing  $\alpha$

The theory of  **$\alpha$ -convex hulls** is necessary to describe the archiving strategy of the  $\Delta_p$ -EMOA that requires the reconstruction and boundary detection of arbitrary shaped point sets in two-dimensional space. Whereas well understood and efficient algorithms exist for generating the convex hull of sets, the determination of the accurate shape of concave areas of a point set is not straightforward. From a statistical perspective the problem is reflected by the distribution based estimation of an unknown set  $S$  given a random sample  $R^S$  of points from  $S$ . For this purpose we use the theory of  $\alpha$ -convex hulls [15] which is provided in the R-package [16] termed `alphahull` [17]. Basically, the concept of convex hulls is generalized. A set  $S$  is  $\alpha$ -convex if any point in the complement  $S^c$  of  $S$  is separable from  $S$  using an open ball of radius  $\alpha$ . The  $\alpha$ -convex hull then is defined as the smallest  $\alpha$ -convex set which contains  $R^S$ . By varying the parameter  $\alpha$  the degree of considered concavity can be controlled. Figure 1 illustrates the effect of increasing  $\alpha$  for a concave point set. It becomes obvious that the  $\alpha$ -convex hull converges to the convex hull with increasing  $\alpha$ .



### 3.3 Specialized EMOA for Evenly Spaced Pareto Fronts

#### 3.3.1 Variants of NSGA2

The first variant of the NSGA2 is denoted by **NSGA2-Seq**. It is based on the conjecture [18] that a *sequential* update of the crowding distances leads to a more homogeneous distribution of the population than the single determination of the crowding distances of the original NSGA2. The second variant of the NSGA2 is denoted by **NSGA2-2NN**. Here, the secondary selection via sequential update of the crowding distances of **NSGA2-Seq** is replaced by the concept of nearest neighbor distances [19] as described in Sec. 3.2. The third variant of the NSGA2 is denoted by **NSGA2-MNN**. As before, the secondary selection via sequential update of the crowding distances of **NSGA2-Seq** is replaced by the concept of nearest neighbor distances. But now  $M > 2$  nearest neighbors are taken into account [19]. By default,  $M = d$ .

#### 3.3.2 Variants of SMS-EMOA

Each of the three variants of the **SMS-EMOA** builds and deploys the  $\Delta_1$ -archive that is detailed below in the pseudo code update (see Alg. 2) and a graphically illustrated example (see Figs. 2 & 3). For all variants the first line of the pseudo code of the original **SMS-EMOA** (see Alg. 1) is extended by initializing the archive set  $A = \text{ND}_f(P, \preceq)$ . The archive update is placed between lines 3 & 4 of the pseudo code in Alg. 1. Variant **SMS<sup>A</sup>-EMOA** uses the archive only for sampling offspring, i.e., it generates offspring alternating from the population  $P$  and the archive  $A$ , which must be reflected in line 3 of the pseudo code in Alg. 1. The population  $P$  is regarded as the approximation of the Pareto front  $F^*$ . If the archive set  $A$  is regarded as the approximation of  $F^*$  then the same variant is termed  $\Delta_p$ -EMOA. If  $A$  is considered as approximation of  $F^*$  but the sampling of offspring only uses  $P$  as parents, then this version is denoted by  $\Delta_p^{\text{pop}}$ -EMOA. The variant in [2] for bi-objective problems with a different archive update was of the latter type.

The concept of the archive update procedure is given in Alg. 2 and visualized in Figs. 2 and 3 in a step-wise manner. The core archive update only comes into effect in case the number of nondominated solutions of the union of the archive  $A$  and the generated candidate point  $x$  exceeds the maximum cardinality  $N_A$  of the archive (line 2). An example of a starting point of the update procedure is shown in Fig. 2.

The current population  $P$  together with the archive  $A$ , constructed by the union of the former  $A$  and the new candidate point  $x$  generated by the SMS-EMOA, is mapped into two-dimensional space in order to circumvent the construction of a local equally spaced reference front in three dimensional space (Fig. 3(a)). Metric MDS is used for this purpose (line 3) as the 3D distance structure is preserved as accurately as possible. The resulting multi-sets are denoted as  $\tilde{P}$  and  $\tilde{A}$ . In case the archive solutions were equally spaced in 3D the respective two-dimensional mapping  $\tilde{A}$  would exhibit a grid structure. As the optimal 2D-Grid consisting of  $\mu$  points and its required rotation is not known a dense grid within the boundary represented by the population  $\tilde{P}$  is used as a substitute. In this step the focus of the

**Algorithm 2** Pseudo code of the MDS-based archive's update procedure

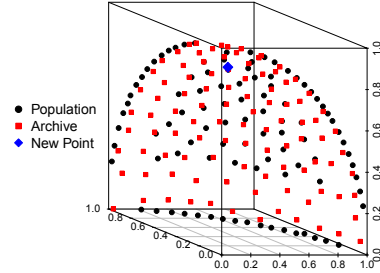
---

```

update( $x, A, P$ ) =
1:  $A = \text{ND}_f(A \cup \{x\}, \preceq)$ 
2: if  $\text{card}(A) > N_A$  then
3:    $(\tilde{A}, \tilde{P}) = \text{MDS}(A, P)$ 
4:    $\tilde{G} = \text{grid}(\tilde{A} \cup \tilde{P}, \delta)$ 
5:    $\tilde{H}_\alpha = \alpha\text{-hull}(\tilde{P})$ 
6:    $\tilde{R} = (\tilde{G} \cap \tilde{H}_\alpha) \cup \partial\tilde{H}_\alpha$ 
7:   for all  $\tilde{a} \in \tilde{A}$  do
8:      $h(\tilde{a}) = \Delta_1(\tilde{A} \setminus \{\tilde{a}\}, \tilde{R})$ 
9:   end for
10:   $\tilde{a}^* = \text{argmin}\{h(\tilde{a}) : \tilde{a} \in \tilde{A}\}$ 
11:   $a^* = \text{indexmap}(\tilde{a}^*)$ 
12:   $A = A \setminus \{a^*\}$ 
13: end if

```

---



**Fig. 2** Pseudo code of the MDS-based archive's update procedure

SMS-EMOA on the PF boundaries is exploited so that it can be assumed that all mapped archive solutions are located inside this two-dimensional boundary.

This grid functions as the reference front w.r.t. which the  $\Delta_p$ -indicator of a subset of  $N_A$  archive solutions has to be minimized. A sequential procedure is carried out in order to construct the grid inside the population boundary. At first, a grid with resolution  $\delta$  within the bounding box of  $\tilde{P}$  is constructed (line 4, Fig. 3 (b)). Next, the  $\alpha$ -convex hull [15]  $\tilde{H}_\alpha$  of  $\tilde{P}$  is determined (line 5, Fig. 3 (c)) and the intersection of the grid points with  $\tilde{H}_\alpha$  is selected. The points of the boundary  $\partial\tilde{H}_\alpha$  of  $\tilde{H}_\alpha$  are added in order to accurately capture the shape of  $\tilde{H}_\alpha$  (line 6, Figs. 3 (d,e)).

The specific archive point  $\tilde{a}^*$  with highest contribution to overall  $\Delta_p$  has to be discarded from the archive: For each point  $\tilde{a}$  within  $\tilde{A}$  the value  $\Delta_p(\tilde{A} \setminus \{\tilde{a}\})$  is computed, and  $\tilde{a}^*$  is determined such that  $\Delta_p(\tilde{A} \setminus \{\tilde{a}^*\})$  is minimal among all  $\Delta_p(\tilde{A} \setminus \{\tilde{a}\})$  (lines 7-9, Fig. 3 (f)). Finally, the index of  $\tilde{a}^*$  is used for removing the corresponding original point in 3D (lines 10, 11).

## 4 Experiments

We experimentally studied the performance of classical EMOA as well as specialized EMOA variants for aiming at equally spaced PF approximations. Comparisons are conducted based on tracking the  $\Delta_p$  indicator in the course of the algorithm runs while the indicator is calculated based on fixed reference fronts referred to as benchmark fronts in the following. Ideal benchmark fronts are composed of the set of  $\mu$  points with minimum  $\Delta_p$  value w.r.t. the true PF with  $\mu$  denoting the population size of the EMOA. However, the construction of these fronts is not straightforward and a matter of current research. Therefore, as the true PFs of the test problems are known in this study, these fronts are densely sampled using approximately

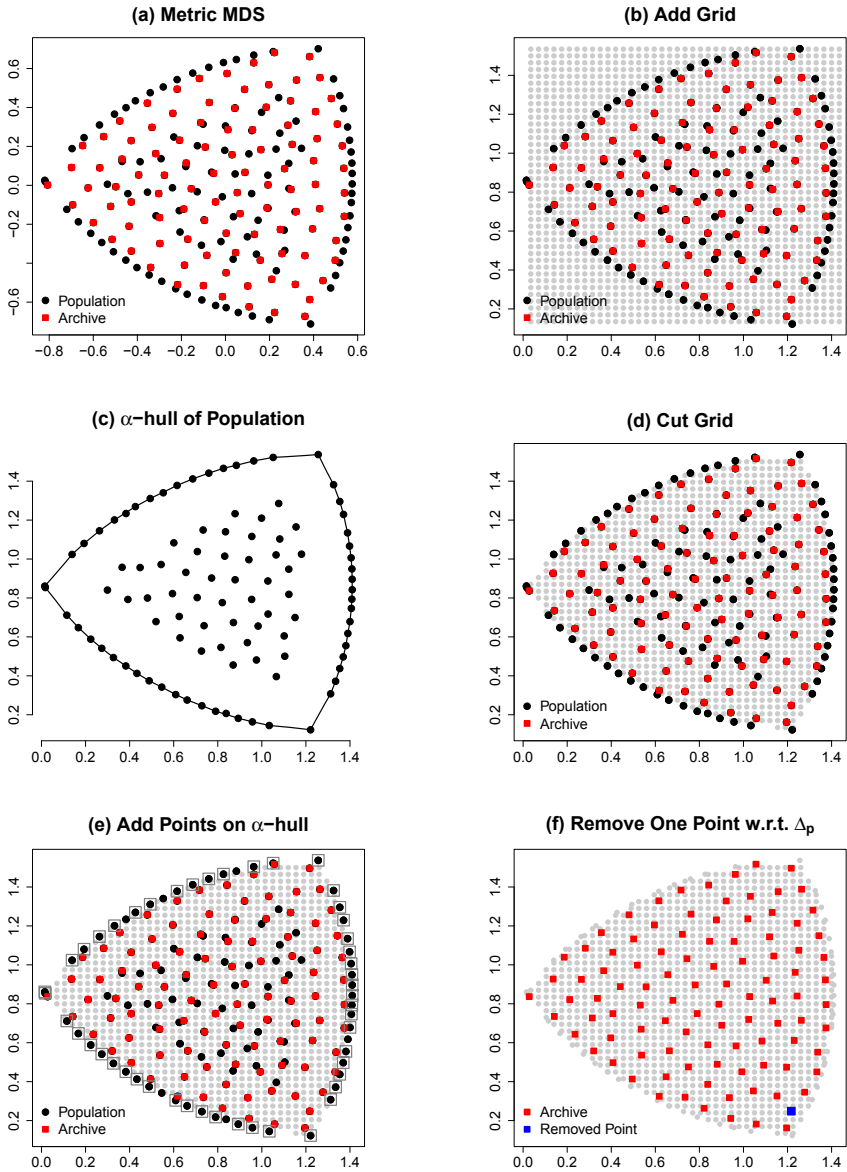


Fig. 3 DTLZ2: Example of Archive Update

10.000 points. As the required benchmark front most probably will be a subset of the latter, Proposition 7 taken from [20] justifies the usage of the densely sampled fronts as a suitable replacement for the required optimal benchmark fronts.

## 4.1 Setup

We consider four standard three-dimensional test problems covering different characteristics of multi-objective optimization problems w.r.t. the shape of the true PF as well as multimodality. The test function set is composed of DTLZ1 (multimodal, linear PF), DTLZ2 (unimodal, concave PF), DTLZ3 (multimodal, concave PF) [20] and the Viennet problem (convex PF) [21]. Decision space dimensions and variable ranges are specified as recommended in literature, i.e.  $0 \leq x_i \leq 1$  ( $i = 1, \dots, 7$ ) for DTLZ1,  $0 \leq x_i \leq 1$  ( $i = 1, \dots, 12$ ) for DTLZ2 /DTLZ3 and  $-2 \leq x_i \leq 2$  ( $i = 1, 2$ ) for Viennet.

NSGA2, NSGA2-Seq, NSGA2-2NN, NSGA2-MNN, MOEA/D, SMS-EMOA as well as the  $\Delta_p$ -EMOA and the  $\Delta_p^{pop}$ -EMOA are applied using a population size of  $\mu = 100$  and 100.000 FE (DTLZ1, DTLZ3), 150.000 FE (Viennet) resp. 90.000 FE for DTLZ2 as EMOA performance stabilizes quite early in the latter case. The  $\Delta_p$ -EMOA,  $\Delta_p^{pop}$ -EMOA, the SMS-EMOA and NSGA2 are implemented in R [16] using the methods of the `emoa` [22] and `mco` [23] package. Furthermore, a MATLAB implementation of MOEA/D is used whereas all NSGA2 variants but the original NSGA2 are integrated into the jMetal framework [24]. Results are also shown for the SMS<sup>A</sup>-EMOA which is an integral part of the  $\Delta_p$ -EMOA as described in Sec. 3.3.

In order to account for the randomness within the EMOA, 5 repeated runs are performed for each setup. The simulated binary crossover operator within the SMS-EMOA and all NSGA2 variants is parameterized by a component-wise probability of  $p_c = 0.9$  and a distribution index of  $\eta_c = 15$  resp.  $\eta_c = 20$  for NSGA2-Seq, NSGA2-2NN and NSGA2-MNN. Polynomial mutation is applied using  $p_m = 1/n$  and the distribution index  $\eta_m = 20$ . The MOEA/D uses Tchebycheff decomposition, its neighborhood parameter equals 10, the subproblem size is set to 100 while the differential evolution parameters  $F$  and  $CR$  are chosen as 0.5.

A value of  $p = 1$  is chosen within the  $\Delta_p$ -EMOA in order to give minimum possible weight to outliers in PF approximations (see [20]). Furthermore, this setting led to best performance results in the experimental investigations of the  $\Delta_p$ -EMOA in 2D [2]. The maximum archive sizes of the  $\Delta_p$ -EMOA, the  $\Delta_p^{pop}$ -EMOA and the SMS<sup>A</sup>-EMOA are set to  $N_A = \mu$ . The parametrization of the  $\Delta_p$ -EMOA archiver is chosen based on preliminary experiments accounting for robustness w.r.t. arbitrary shapes of the point set resulting from MDS which led to  $\alpha = 5$ . In addition, a sufficiently high number of grid points inside the  $\alpha$ -convex hull is required for detecting the point with the lowest contribution to an improvement of  $\Delta_p$ . The grid solution is set to 50 (DTLZ1, Viennet) resp. 40 for (DTLZ2, DTLZ3) in order to keep the

computational effort at a reasonable level. In the latter cases a much higher number of grid points falls inside the  $\alpha$ -convex hull of the point set due its convex shape.

## 4.2 Results

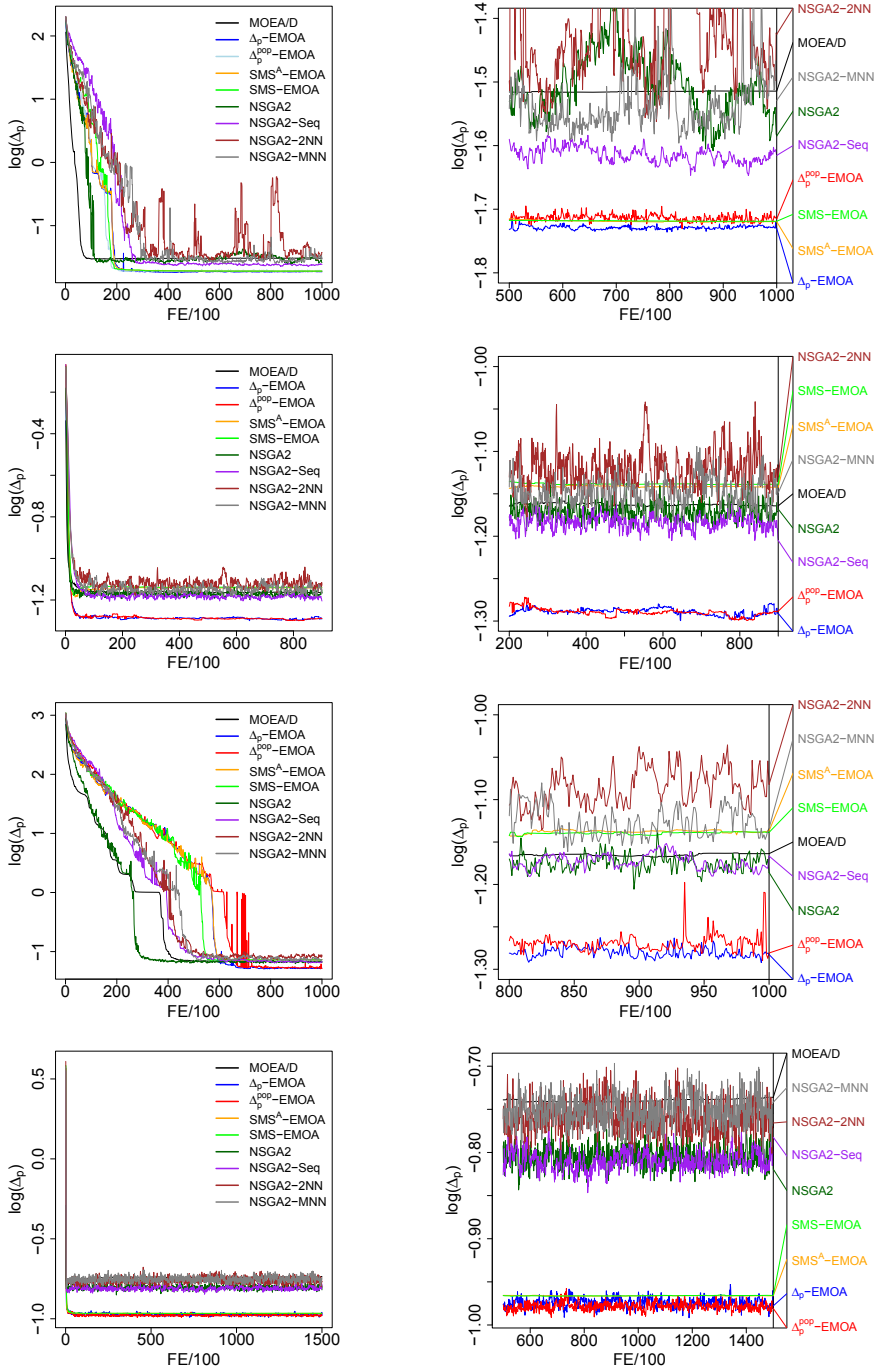
Figure 4 visualizes the median  $\Delta_p$  values in the course of the algorithm runs for all considered algorithm and test problem combinations. It becomes obvious that the  $\Delta_p$ -EMOA outperforms the remaining EMOA variants but the  $\Delta_p^{pop}$ -EMOA w.r.t.  $\Delta_p$ ; if not directly from the start this effect occurs after an initial phase. The amount of improvement is much higher for the concave than for the convex problems. For both multimodal problems DTLZ1 and DTLZ3 the challenge of escaping from local subfronts becomes visible which is handled best by the SMS-EMOA and MOEA/D. In addition, the effect of the alternating sampling from population and archive for the parent selection within the proposed algorithm can be investigated as well. No definite conclusion can be drawn in this regard as the ranking of the  $\Delta_p$ -EMOA and the  $\Delta_p^{pop}$ -EMOA on the test problems is not clear without ambiguity.

However, a special situation is given for DTLZ1 due to the linear shape of its true PF. In this case the optimal distribution of solutions w.r.t. the dominated hypervolume is composed of equally spaced points. Therefore, the SMS-EMOA directly aims at equal spacing already and shows similar performance to the  $\Delta_p$ -EMOA w.r.t.  $\Delta_p$ . Furthermore, this is the only test problem for which the  $\Delta_p$ -EMOA clearly shows superior performance regarding the  $\Delta_p^{pop}$ -EMOA.

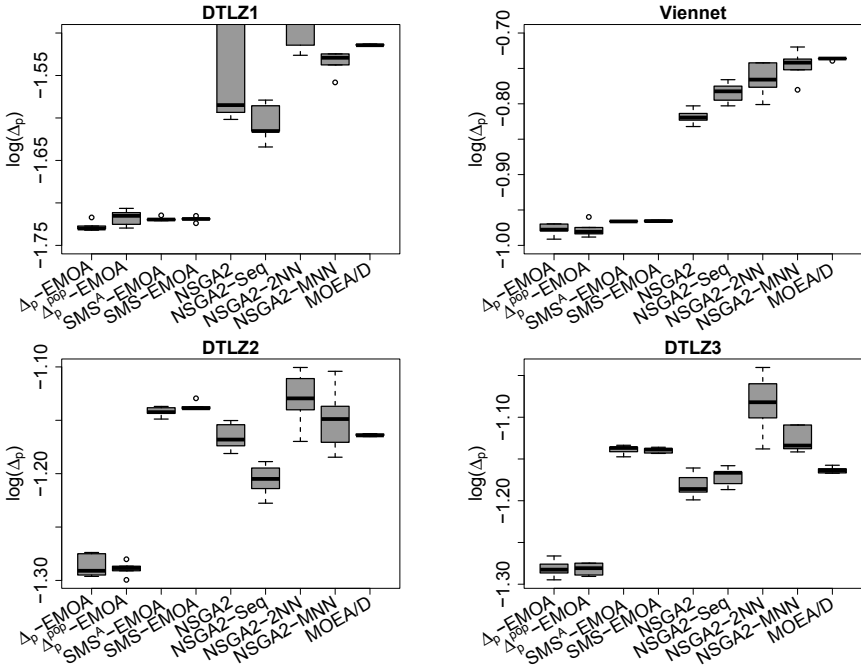
In order to account for the stochastic nature of the EMOA and thus to be sure that performance differences are significant, Figure 5 shows boxplots of the  $\Delta_p$ -indicator at the final generation. It becomes obvious that the differences in location of the  $\Delta_p$ -values of the  $\Delta_p$ -EMOA compared to all other EMOA variants but the  $\Delta_p^{pop}$ -EMOA are statistically significant. As indicated above, performance differences are much higher for DTLZ2 and DTLZ3. However, the performance of the  $\Delta_p$ -EMOA and the  $\Delta_p^{pop}$ -EMOA cannot be distinguished from each other for DTLZ2, DTLZ3 and Viennet.

Figures 4 and 5 reveal that MOEA/D is not competitive w.r.t.  $\Delta_p$  on all test problems. The same is true for NSGA2 as well as the respective specialized variants. While the sequential crowding variant NSGA2-Seq performs best within this EMOA subgroup, NSGA2-2NN and NSGA2-MNN surprisingly show much worse performance than the original NSGA2.

Certainly, the whole performance comparison would change if the HV indicator was used as a quality indicator as equally spaced solutions do not result in optimal HV values. The optimal distribution is rather biased towards the boundary and the knee points of the front. This is reflected in Figure 6 which visualizes the best PF approximations out of the repeated runs obtained w.r.t.  $\Delta_p$  for selected EMOA and test problems. As the resulting PF approximations for DTLZ3 are very similar to the respective ones of DTLZ2, the former are omitted. The  $\Delta_p$ -EMOA is contrasted to SMS-EMOA and MOEA/D. NSGA2-Seq is chosen from the set of NSGA2-variants as it turned out to be the most competitive EMOA within this set.



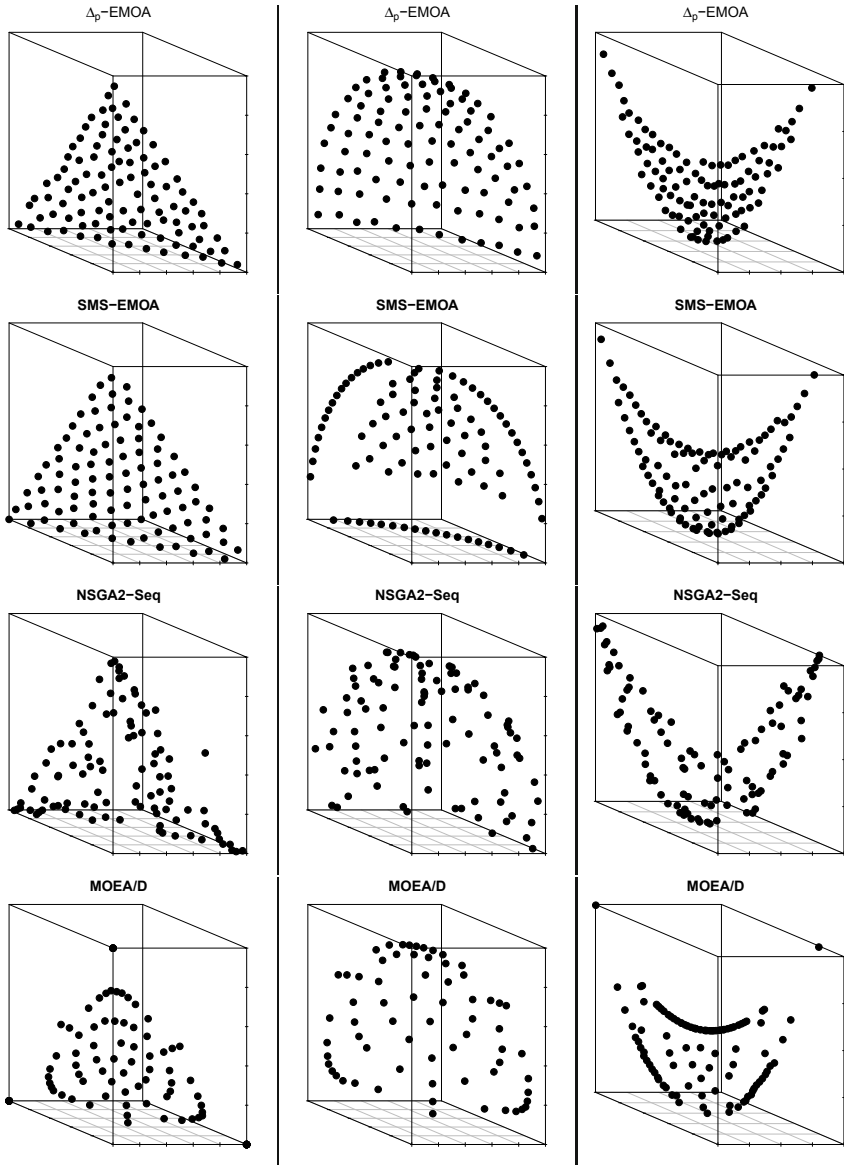
**Fig. 4** Performance results of the considered algorithms w.r.t.  $\Delta_p$  for DTLZ1 (top), DTLZ2 (middle top), DTLZ3 (middle bottom) and Viennet (bottom)



**Fig. 5** Boxplots of  $\Delta_p$ -values at final generation (100000 FE (DTLZ1,DTLZ3), 90000 FE (DTLZ2), 150000 FE (Viennet))

Performance differences are clearly reflected in the visualizations of the PF approximations. Whereas the SMS-EMOA quite accurately approximates the border solutions of the true PF and focusses on knee points, the  $\Delta_p$ -EMOA quite successfully manages to generate evenly spaced solutions. In the special case of DTLZ1 both approaches rather coincide due to the reasons stated above. The PF approximations of NSGA2-Seq and MOEA/D both exhibit gaps in an unstructured manner.

Finally, we must state some notes on the methods's computational complexity: The major goal of this work was an extension of our original method from 2 to 3 objective functions. Since the approach of extending the method was not obvious several ideas have been evaluated in a kind of rapid prototyping. According to the well known programmer's principle "Make it run first, then make it run fast" we initially favored effectivity (the method produces the intended result) over efficiency (the method performs in best possible manner with least waste of resources). Therefore the current implementation of our method (in R) is slow but effective. The analysis of its computational complexity and a fast re-implementation are the next steps of our agenda.



**Fig. 6** Best PF approximations of selected algorithms w.r.t.  $\Delta_p$  for DTLZ1 (left, 100000 FE), DTLZ2 (middle, 90000FE) and Viennet (right, 150000 FE). Results for DTLZ3 are very similar to DTLZ2.



## 5 Conclusions

The experiments have shown that the proposed  $\Delta_p$ -EMOA for three objectives is able to generate evenly spread PF approximations for the considered test problems. In addition, it outperforms standard EMOA such as NSGA2, SMS-EMOA and MOEA/D as well as specialized variants of NSGA2 for this purpose. In the special situation of linear PFs the performance of the  $\Delta_p$ -EMOA is comparable to the SMS-EMOA as the optimal placement of points on a linear hyperplane regarding the HV coincides with equally spaced points. However, no significant effect of alternated sampling from population and archive within the proposed EMOA could be observed.

In future studies we will focus on reducing the computational complexity of the  $\Delta_p$ -EMOA archiving strategy in order to keep the algorithm runtimes at a reasonable level. This includes parallelization approaches and strategies for the optimal number and placement of the grid points in the archive update strategy. Furthermore, alternatives for the task of constructing concave hulls will be considered. Furthermore, it will be investigated if the performance of the  $\Delta_p$ -EMOA can be significantly improved by tuning the internal mutation and variation operators.

Moreover,  $\mu$ -optimal distributions of solutions w.r.t.  $\Delta_p$  will be analyzed theoretically. On the other hand, effort will be spent in selecting and tuning suitable optimizers for approximating the latter as these sets are to serve as reference fronts for EMOA benchmarking w.r.t.  $\Delta_p$ .

In addition, we will investigate the  $\Delta_p$ -algorithm behavior for higher objective space dimensions as the underlying concept is not restricted to three-dimensional problems. The computational complexity of the archiving strategy will roughly stay the same due to the mapping to two-dimensional space. The only factor to be considered is the computation of the distance matrix. However, it has to be kept in mind that the complexity of the HV computation increases exponentially with increasing dimension so that the SMS-EMOA will require a considerably higher amount of runtime.

**Acknowledgements.** The authors acknowledge support from DAAD project no. 51222288, CONACYT project no. 146776 and 174443 as well as from the DFG, project no. TR 891/5-1.

## References

1. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* (2012), doi:10.1109/TEVC.2011.2161872
2. Gerstl, K., Rudolph, G., Schütze, O., Trautmann, H.: Finding evenly spaced fronts for multiobjective control via averaging hausdorff-measure. In: *Int'l. Proc. Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2011)*, pp. 975–980 (2011)

3. Mehnen, J., Wagner, T., Rudolph, G.: Evolutionary optimization of dynamic multi-objective test functions. In: Proceedings of the Second Italian Workshop on Evolutionary Computation, GSICE2 (2006); Published on CD-ROM
4. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181(3), 1653–1669 (2007)
5. Pareto, V.: *Manual of Political Economy*. The MacMillan Press (1971)
6. Knowles, J., Corne, D.: On metrics for comparing nondominated sets. In: Congress on Evolutionary Computation (CEC 2002), vol. 1, pp. 711–716. IEEE Press, Piscataway (2002)
7. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2), 117–132 (2003)
8. Van Veldhuizen, D.A.: *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1999)
9. Coello Coello, C.A., Cruz Cortés, N.: Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines* 6(2), 163–190 (2005)
10. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
11. Beume, N.: S-metric calculation by considering dominated hypervolume as Klee's measure problem. *Evolutionary Computation* 17(4), 477–492 (2009)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
13. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6), 712–731 (2007)
14. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer, New York (2001)
15. Pateiro-Lopez, B., Rodriguez-Casal, A.: Generalizing the convex hull of a sample: The R package alphahull. *Journal of Statistical Software* 34(5), 1–28 (2010)
16. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2011), ISBN 3-900051-07-0
17. Pateiro-Lopez, B., Rodriguez-Casal, A.: alphahull: Generalization of the convex hull of a sample of points in the plane. CRAN (2011), <http://cran.r-project.org/web/packages/alphahull/index.html>
18. Kukkonen, S., Deb, K.: Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In: Proc. Congress on Evolutionary Computation (CEC 2006), vol. 1, pp. 1179–1186. IEEE Press, Piscataway (2006)
19. Kukkonen, S., Deb, K.: A fast and effective method for pruning of non-dominated solutions in many-objective problems. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 553–562. Springer, Heidelberg (2006)
20. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Fogel, D., et al. (eds.) Proc. Int'l. Congress on Evolutionary Computation (CEC 2002), vol. 1, pp. 825–830. IEEE press (2002)

21. Viennet, R., Fonteix, C., Marc, I.: Multicriteria optimization using a genetic algorithm for determining a pareto set. *International Journal of Systems Science* 27(2), 255–260 (1996)
22. Mersmann, O.: emoa: Evolutionary Multiobjective Optimization Algorithms, R package version 0.4-8 (2012), <http://cran.r-project.org/web/packages/emoa/>
23. Mersmann, O., Trautmann, H., Steuer, D.: mco: Multi criteria optimization algorithms and related functions, R package version 1.0.9 (2012), <http://cran.r-project.org/web/packages/mco/>
24. Durillo, J.J., Nebro, A.J.: jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* 42, 760–771 (2011)

# Software Requirements Optimization Using Multi-Objective Quantum-Inspired Hybrid Differential Evolution

A. Charan Kumari, K. Srinivas, and M.P. Gupta

**Abstract.** Multi-Objective Next Release Problem (MONRP) is an important software requirements optimization problem in Search-Based Software Engineering. As the customer requirements varies from time to time, often software products are required to incorporate these changes. It is a hard task to optimize the requirements from a large number of candidates, for the accomplishment of the business goals and at the same time, the satisfaction of the customers. MONRP identifies a set of requirements to be included in the next release of the product, by minimizing the cost in terms of money or resources, and maximizing the number of customers to get satisfied by including these requirements. The problem is multi-objective in nature and the objectives are conflicting objectives. The problem is NP-hard and since it cannot be solved effectively and efficiently by traditional optimization techniques especially for large problem instances, Metaheuristic Search and Optimization Techniques are required. Since MONRP has wide applicability in software companies and manufacturing companies, there is a need for efficient solution techniques especially for the large problem instances. Therefore, this paper presents a Multi-objective Quantum-inspired Hybrid Differential Evolution (MQHDE) for the solution of MONRP which combines the strengths of Quantum Computing, Differential Evolution and Genetic Algorithm. The features of MQHDE help in achieving

---

A. Charan Kumari

Research scholar, Department of Physics and Computer Science,  
Dayalbagh Educational Institute, Agra, India  
e-mail: [charankumari@yahoo.co.in](mailto:charankumari@yahoo.co.in)

K. Srinivas

Department of Electrical Engineering, Dayalbagh Educational Institute, Agra, India  
e-mail: [ksri12@gmail.com](mailto:ksri12@gmail.com)

M.P. Gupta

Department of Management Studies, Indian Institute of Technology, Delhi, India  
e-mail: [mpgupta@dms.iitd.ernet.in](mailto:mpgupta@dms.iitd.ernet.in)

consistency in performance in terms of convergence to Pareto-optimal front, good spread among the obtained Pareto-optimal front solutions, faster convergence and obtaining relatively more number of solutions along the Pareto-optimal front. The performance of MQHDE is tested on six benchmark problems using Spread and HyperVolume metrics. The comparison of the obtained results indicates consistent and superior performance of MQHDE over the other methods reported in the literature.

**Keywords:** Search-based software engineering, Multi-objective optimization, Multi-objective Quantum-inspired Hybrid Differential Evolution, Multi-objective next release problem.

## 1 Introduction

Companies involved in developing and maintaining large complex software systems, require to determine the requirements of the customers to be included into its next release. This problem has been formulated as Next Release Problem (NRP) by Bagnall et al. [1] and is widely referenced by researchers in the field of search-based software engineering. The problem is defined as to identify a set of requirements to be included into the next release of the software product, by satisfying the demands of the customers to the maximum extent, and at the same time ensuring the minimum utilization of the resources as far as possible. The goal of NRP is to balance customer satisfaction, resource constraints and requirement dependencies. They applied various techniques including Greedy algorithms and simulated annealing on a set of five randomly generated data sets with increasing dimensionality of the problem.

To model a more realistic and real life application, the NRP has been formulated as Multi-Objective Next Release Problem (MONRP). As, minimizing the cost and maximizing the customer satisfaction, are two contradictory objectives, the NRP, has been formulated as Multi-Objective Next Release Problem (MONRP) by Zhang et al. [2]. Later, Durillo. J. et al.[3] studied MONRP by applying NSGA II, Pareto GA, Single-objective GA and Random Search on a set of six randomly generated data sets and experimented with the scaling and boundary issues of the MONRP.

In this paper, we present a Multi-objective Quantum-inspired Hybrid Differential Evolution (MQHDE) for the solution of the Multi-Objective Next Release problem. The basic model of MONRP is considered in this work due to its wide applicability in all software companies and manufacturing companies. Though the metaheuristic search techniques do not guarantee to provide optimal solutions, yet, they can obtain near-optimal solutions in a reasonable amount of computational time. Thus, the solution to a real-world large-scale MONRP needs further exploration and experimentation with faster and more efficient metaheuristics. MQHDE combines the strengths of Quantum Computing, Differential Evolution and Genetic Algorithms. This hybridization leads to the achievement of the right balance in exploration and exploitation of the search space on one hand and the effective handling of the two conflicting goals (convergence and diversity) of Multi-objective optimization on the other hand. The features of MQHDE help in achieving consistency in performance

in terms of convergence to Pareto-optimal front, good spread among the obtained Pareto-optimal front solutions, faster convergence and obtaining relatively more number of solutions along the Pareto-optimal front. The performance of MQHDE algorithm was studied on the six data sets proposed in [2, 3], and compared with the reported results [3].

The rest of the paper is structured as follows. Section 2 describes the main concepts of Multi-objective optimization. The formulation of software requirements optimization process as Multi-Objective Next release Problem is presented in Section 3. The pseudo code of MQHDE along with the detailed description of the concepts is given in Section 4. Section 5 presents the experiments. The results obtained by the algorithm and an analysis of the results are shown in Section 6. Finally, concluding remarks are given in Section 7.

## 2 Multi-Objective Optimization

The problems with two or more objective functions are called multi-objective and require special mathematical and algorithmic tools for optimization. Most of the real world problems are multi-objective in nature. In multi-objective optimization, the notion of optimality changes as these solutions are trade-offs or good compromises among the objectives [4]. In order to generate these trade-off solutions, a notion of optimality called Edgeworth-Pareto optimality is used which states that a solution to a Multi-objective problem is Pareto optimal, if there exists no other feasible solution which would improve some criterion without causing a simultaneous degrading in at least one other criterion. The use of this concept almost always gives a set of non-dominated solutions, which is called the Pareto optimal set.

The remainder of this section gives formal definitions of the above mentioned concepts. [5].

### 2.1 Multi-Objective Optimization Problem (MOP)

A general MOP is defined as minimizing (or maximizing)  $F(x) = (f_1(x), f_2(x), \dots, f_k(x))$  subject to  $g_i(x) \leq 0$ ,  $i = 1, 2, \dots, m$  and  $h_j(x) = 0$ ,  $j = 1, 2, \dots, p$  and  $x \in \Omega$ . A MOP solution minimizes (or maximizes) the components of a vector  $F(x)$  where  $x$  is a  $n$ -dimensional decision variable vector  $x = (x_1, x_2, \dots, x_n)$  from some universe  $\Omega$ ; and  $g_i(x) \leq 0$  and  $h_j(x) = 0$  represent constraints that must be fulfilled while minimizing (or maximizing)  $F(x)$  and  $\Omega$  contains all possible  $x$  that can be used to satisfy an evaluation of  $F(x)$ .

Thus, a MOP consists of  $k$  objectives represented as  $k$  objective functions with  $m$  inequality and  $p$  equality constraints on the objective functions and  $n$  decision variables. The evaluation function  $F$  is a mapping from the vector of decision variables to output vectors. The output vector which satisfies  $(m + p)$  constraints is known as a feasible solution and the set of all feasible solutions constitutes the feasible region.

## 2.2 Pareto Dominance

A vector  $u = (u_1, u_2, \dots, u_k)$  is said to dominate another vector  $v = (v_1, v_2, \dots, v_k)$  if and only if  $u$  is partially less than  $v$ , i.e.,  $\forall i \in \{1, 2, \dots, k\}, u_i \leq v_i$  and  $\exists i \in \{1, 2, \dots, k\} : u_i < v_i$ .

A solution is said to Pareto dominate another, if the first solution is not inferior to the second solution in all objectives, and there is at least one objective where it is better.

## 2.3 Pareto Optimality

A solution  $x \in \Omega$  is said to be Pareto optimal with respect to  $\Omega$  if and only if there is no  $x' \in \Omega$  for which  $v = F(x') = (f_1(x'), f_2(x'), \dots, f_k(x'))$  dominates  $u = F(x) = (f_1(x), f_2(x), \dots, f_k(x))$ .

In other words, the solution  $x^*$  is called Pareto optimal, if there exists no other feasible solution  $x$ , which would decrease some criterion without causing a simultaneous increase in at least one other criterion.

## 2.4 Pareto Optimal Set

For a given MOP  $F(x)$ , the Pareto Optimal Set  $P^*$  is defined as:

$$P^* = \{x \in \Omega \mid \neg \exists x' \in \Omega F(x') \preceq F(x)\}.$$

The Pareto optimal set consists of all solutions that satisfy the condition of Pareto dominance.

## 2.5 Pareto Front

For a given MOP  $F(x)$  and Pareto Optimal Set  $P^*$ , the Pareto Front  $PF^*$  is defined as:

$$PF^* = \{F(x) \mid x \in P^*\}.$$

Pareto front is obtained when the Pareto optimal set is plotted on an objective space.

In Multi-Objective optimization (MOO), the two distinct goals are progressing towards the Pareto-optimal front and maintaining a diverse set of solutions in the front. Since both goals are important, an effective MOO algorithm must achieve both of them within a reasonable computational effort.

### 3 Multi-Objective Next Release Problem

This section describes the software requirements selection process as Multi-objective Next Release Problem as formulated by Zhang et al. [2].

Given an existing software package, there is a set  $C$  consisting of  $m$  customers.  $C = \{c_1, c_2, c_3, \dots, c_m\}$ , whose requirements are to be considered for the next release of the product. The set of requirements proposed by the customers for the next release is denoted by  $R = \{r_1, r_2, r_3, \dots, r_n\}$ . Each customer has a degree of importance for the company based on factors such as consistency in orders, payment terms, credibility etc. that can be revealed by a weight factor. The set of relative weights associated with each customer  $c_j$  ( $1 \leq j \leq m$ ) is denoted by  $Weight = \{w_1, w_2, w_3, \dots, w_m\}$ , where  $w_j \in [0,1]$  and  $\sum_{j=1}^m w_j = 1$ .

For implementing each requirement, resources such as manpower, hardware and software tools are needed, which can be translated in terms of cost. The cost associated with each requirement  $r_i$  ( $1 \leq i \leq n$ ) for its implementation is designated by  $Cost = \{cost_1, cost_2, cost_3, \dots, cost_n\}$ . As all the requirements are not equally important for the customers, each customer  $c_j$  ( $1 \leq j \leq m$ ) assigns a value for requirement  $r_i$  ( $1 = i = n$ ), denoted by value  $(r_i, c_j)$ .

The score of requirement  $r_i$  can be calculated as

$$score_i = \sum_{j=1}^m w_j * value(r_i, c_j) \quad (1)$$

The decision vector  $x = \{x_1, x_2, x_3, \dots, x_n\} \in \{0,1\}$  is a solution vector that indicates the requirements that are to be included in the next release of the product. In this vector  $x_i = 1$ ; if the requirement is selected for inclusion in the next release and 0, otherwise.

The two objectives to optimize can be formulated as

$$\text{minimize } f_1 = \sum_{i=1}^n cost_i * x_i \quad (2)$$

$$\text{maximize } f_2 = \sum_{i=1}^n score_i * x_i \quad (3)$$

Since minimizing the function  $f_1$  is same as maximizing the function  $-f_1$ ; In this paper, we have taken the maximization of  $-f_1$  (cost) and maximization of  $f_2$  (customer satisfaction).

The main goal in implementing the MONRP is to find a set of requirements that are to be included in the next release of the software product by minimizing the cost and at the same time by maximizing the customer satisfaction.



## 4 Multi-Objective Quantum-Inspired Hybrid Differential Evolution

Differential evolution (DE) is a stochastic, population based evolutionary algorithm [6] widely used for solving multi dimensional global optimization problems over continuous spaces. However, the design of its operators makes it unsuitable for many combinatorial optimization problems which operate on binary space. On the other hand, Quantum-inspired Evolutionary Algorithms (QEAs) [7] have demonstrated better performance over other evolutionary techniques in the attempted problems [8] due to their ability to balance exploration and exploitation of the search space, robust search and efficiency in representation scheme. They apply the Quantum Computing principles to enhance the capabilities of classical evolutionary algorithms. In this paper, we present a Multi-objective Quantum-inspired Hybrid Differential Evolution (MQHDE) that extends the concepts of differential and genetic operators to the quantum paradigm to fit into the multi-objective frame work. This section describes the framework of the MQHDE. The pseudo code of MQHDE is presented as Algorithm 1.

### 4.1 Representation

MQHDE uses a probabilistic representation that is based on the concepts of qubits. It maintains a quantum population  $Q = (q_1, q_2, q_3, \dots, q_n)$  of size 'n', where  $q_i$  is a qubit individual defined as

$$q_i = (\gamma_{i1}, \gamma_{i2}, \gamma_{i3}, \dots, \gamma_{im}) \quad (4)$$

where  $m$  is the length of the qubit individual.  $|\gamma_{ij}|^2$  represents the probability that qubit is in state 0 and  $1 - |\gamma_{ij}|^2$  represents the probability that the qubit is in state 1. Each  $\gamma_{ij}$  is initialized in the range  $[0, \frac{1}{\sqrt{2}}]$ .

In the process of observation, each qubit individual  $q_i$  is observed to make a binary solution  $P_i$ , using the following pseudo code.

```

procedure Make(q)
begin
    j = 1
    while (j < m)
    {
        if (rand[0,1] >  $|\gamma_{ij}|^2$ )
             $p_{ij} = 1$ 
        else
             $p_{ij} = 0$ 
        end;
    }

```

```

    }
    j = j + 1
}
end

```

## 4.2 Mutation Operator

The Mutation operator is applied on qubit individuals and can be expressed as [9]

$$q_i^m(t) = q_{r1}(t-1) + F(q_{r2}(t-1) - q_{r3}(t-1)) \quad (5)$$

where  $r1$ ,  $r2$ , and  $r3$  are three random numbers, and are all distinct and also different from the running index  $i$ . The parameter  $t$  denotes the generation number. The negative values obtained during the mutation are treated by the observation process as described in the procedure 'Make'. If the mutated values exceeds the range  $[0,1]$ ; then they are set to the boundary values.  $F$  is the mutation control parameter and is generally set in the range of  $[0,2]$ .  $F$  is initially set to 2 and uniformly decremented over the iterations. The main motive in doing so is to enable maximum exploration of the search space and then later when the search stabilizes, gradually the value of  $F$  is decreased to enable proper exploitation of reached good regions of the search space. During the experimentation it has been observed that this feature of the proposed algorithm further contributed in the improvement of quality of solution.

---

### Algorithm 1 Pseudo code of MQHDE

---

- 1:  $t = 0$
  - 2: Initialize  $Q(t)$  a population of 'N' qubit individuals with 'm' qubits in each.
  - 3: Obtain  $P(t)$  by observing the states of  $Q(t)$ .
  - 4: Evaluate fitness of  $P(t)$ .
  - 5: **while** not termination condition **do**
  - 6:    $t = t + 1$
  - 7:   Obtain the mutant population  $Q_m(t)$  from  $Q(t-1)$  as  $q_m(t) = q_{r1}(t-1) + F(q_{r2}(t-1) - q_{r3}(t-1))$ , where  $r1 \neq r2 \neq r3 \neq i$  and  $F \in [0,2]$ .  $F$  is initially set to 2 and uniformly decremented over all the Iterations. If the value of  $q_m(t)$  goes out of range set it at its corresponding limit.
  - 8:   Obtain  $P_m(t)$  by observing the states of  $Q_m(t)$ .
  - 9:   Evaluate the fitness of  $P_m(t)$ .
  - 10:   Obtain the crossover population  $Q_c(t)$  from  $Q(t-1)$  using single point crossover with a crossover probability of 0.8.
  - 11:   Obtain  $P_c(t)$  by observing the states of  $Q_c(t)$ .
  - 12:   Evaluate the fitness of  $P_c(t)$ .
  - 13:   Perform fast non dominated sort on  $P(t-1) \cup P_m(t) \cup P_c(t)$ .
  - 14:   Form  $Q(t)$  by accommodating distinct quantum individuals pertaining to the different Pareto-fronts starting from the best front by taking crowding distance into consideration.
  - 15: **end while**
-

### 4.3 Crossover Operator

The crossover operation operates on the original Quantum individuals. In MQHDE, a single point crossover is performed with a crossover probability of 0.8. Two Quantum individuals and a crossover point are selected at random and the qubits to the right side of the crossover point are exchanged between the two quantum individuals to produce crossover quantum individuals of the crossover population.

### 4.4 Selection

The qubit individuals of the next generation are selected after performing a fast non dominated sorting algorithm on the population obtained by combining original, mutant and crossover populations. Algorithm 2 contains the pseudo code of fast non dominated sorting algorithm as described by Deb et al. [10]. In this process for each solution two entities are calculated - domination count  $n_p$ , the number of solutions which dominate the solution  $p$  and  $S_p$  a set of solutions that the solution  $p$  dominates. All solutions in the first nondominated front will have their domination count as zero. Then, for each solution  $p$  with  $n_p = 0$ , the domination count of each member ( $q$ ) of its set  $S_p$  is decremented by one. Then the second nondominated front is identified as all  $q$  for which the domination count become zero. The process is repeated until all the fronts are identified.

The quantum population  $Q(t)$  for the next generation is obtained by accommodating the distinct quantum individuals pertaining to different Pareto-fronts starting from the first front. If the number of individuals in the Pareto front being considered is less than or equal to the number of vacant slots available in  $Q(t)$ , then all the solutions of the front are accommodated. Otherwise, required number (equal to the available slots) of individuals are selected from the Pareto front being considered, based on the crowding distance to ensure wide spread of solutions.

## 5 Experiments

This section describes the test problems taken from the literature and the indicators used to measure the performance of the obtained Pareto fronts and also specify the values assigned for various parameters.

### 5.1 Test Problems

The algorithm is tested on a set of six benchmark problems that have been proposed in [2, 3]. There are two test problem sets. In the first set, three ‘scales’ of

**Algorithm 2** Pseudo code of fast non dominated sorting

---

```

1: for all  $p \in P$  do
2:    $S_p = \emptyset$ 
3:    $n_p = 0$ 
4:   for all  $q \in P$  do
5:     if  $p$  dominates  $q$  then
6:        $S_p = S_p \cup \{q\}$ 
7:     else
8:        $n_p = n_p + 1$ 
9:     end if
10:  end for
11:  if  $n_p = 0$  then
12:     $p_{rank} = 1$ 
13:     $F_1 = F_1 \cup \{p\}$ 
14:  end if
15: end for
16:  $i = 1$ 
17: while  $F_i \neq \emptyset$  do
18:    $Q = \emptyset$ 
19:   for all  $p \in F_i$  do
20:     for all  $q \in S_p$  do
21:        $n_q = n_q - 1$ 
22:       if  $n_q = 0$  then
23:          $q_{rank} = i + 1$ 
24:          $Q = Q \cup \{q\}$ 
25:       end if
26:     end for
27:   end for
28:    $i = i + 1$ 
29:    $F_i = Q$ 
30: end while

```

---

problems are considered, with the number of customers ranging from 15 to 100 and the number of requirements ranging from 40 to 140. The second set, is concerned with ‘bounding’ NRP. This benchmark problem set is comprised of three instances: an instance with 20 requirements and 100 customers (few requirements, high number of customers), an instance with 25 requirements and 100 customers, and other with only 2 customers and 200 requirements (high number of requirements, few customers).

## 5.2 Performance Measures

For assessing the quality of the results obtained by the multi-objective algorithms two issues are generally taken into consideration: (i) progressing towards the Pareto-optimal front (convergence) (ii) maintaining a diverse set of solutions in the front

(diversity). A number of metrics have been identified and reported in the literature for assessing the quality of various MOEAs. In this section we have considered two metrics: (a) Spread ( $\Delta$ ) to measure the diversity among the non dominated solutions and (b) Hypervolume for measuring both closeness and diversity.

In order to compute the spread metric, a Pareto optimal front is required. As the optimal Pareto front is not known in this case, a reference Pareto optimal front is constructed by taking the non dominated solutions obtained during 30 independent runs of the MQHDE algorithm.

- **Metric for Diversity:** The spread metric  $\Delta$  [4] measures the extent of spread by the obtained solutions

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q-1|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q-1| \bar{d}} \quad (6)$$

where  $d_i$  can be any distance measure between consecutive solutions and  $\bar{d}$  is the mean value of these distance measures. The parameter  $d_m^e$  is the distance between the extreme solutions of obtained front and optimal Pareto front corresponding to the  $m$ -th objective function. This metric is also applied after normalizing the objective function values. For an ideal distribution, this metric takes a value zero.

- **Metric for Convergence and Diversity:** The Hypervolume (HV) [11] metric calculates the volume covered by the members of the obtained front in the objective space. Mathematically, for each solution  $i \in Q$ , a hypercube  $v_i$  is constructed with a reference point  $w$  and the solution  $i$  as the diagonal corners of the hypercube. The reference point is found by taking the maximum possible objective values so that even the extreme solutions found by the algorithm are included in calculating the hypervolume. The reference point may match with the extreme solutions only if the algorithm selects all the requirements; which is a rare case. And as the objective function values are of differing magnitude, the objective function values are normalized and (1, 1) is taken as the reference point. A union of all the hyper cubes is found and its hypervolume is calculated.

$$HV = volume(\cup_{i=1}^{|Q|} v_i) \quad (7)$$

Solutions with larger values of HV are desirable.

### 5.3 Experimental Setup

In all the test cases, the range of 'cost' is from 1 through 9 inclusive and the range of 'value' is from 0 to 5 inclusive. All the customers are given equal importance. The population size is taken as 100 and all the problems were run for a maximum

of 10000 function evaluations. The algorithm has been implemented in MATLAB 7.6.0, on an Intel®Core™ 2 Duo CPU T6600 @2.20 GHz processor, 3 GB RAM and Windows 7 platform.

## 5.4 Methodology

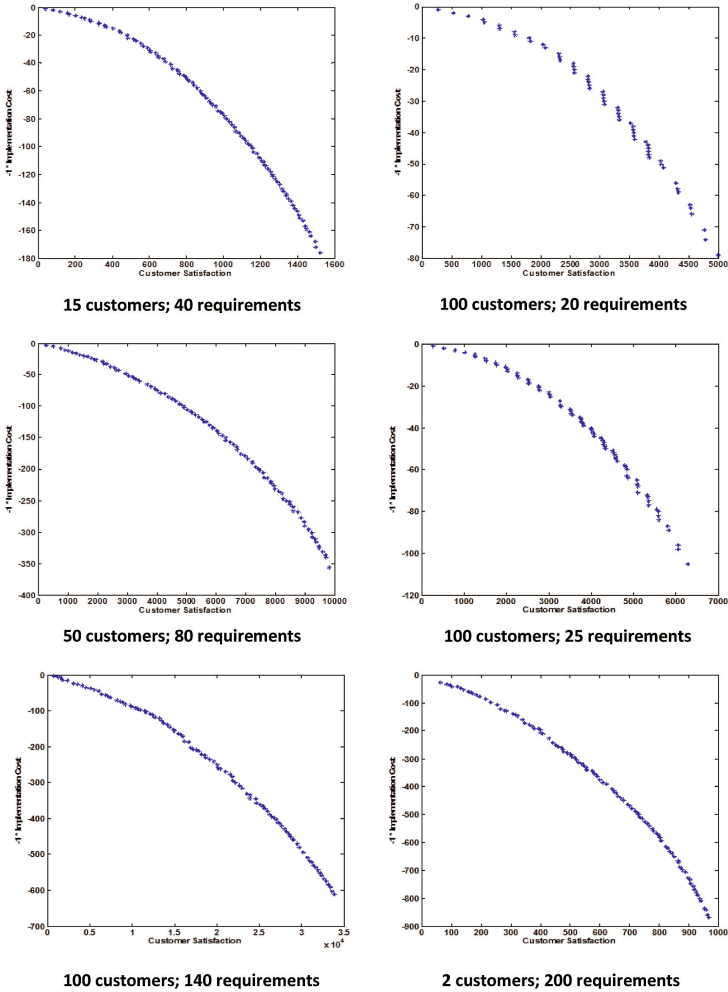
The Pareto fronts obtained with MQHDE for each problem instance are shown. To assess the performance of MQHDE, we have performed 100 independent runs of each problem instance and calculated the mean and standard deviation of the two metrics Spread ( $\Delta$ ) and Hypervolume (HV) for comparison with the reported results of two state-of-the-art multi-objective algorithms NSGA II and MOCell [3].

## 6 Results and Analysis

In this section we present the results obtained by MQHDE. We assess the performance of MQHDE mainly on the basis of obtained quality of solution and results of Spread and Hypervolume metrics. A comparative assessment of the performance is also done with reference to results obtained by the other multi-objective evolutionary algorithms reported in the literature. Though the constructed reference fronts may not exactly match with those reference fronts used in [3] and hence do not result in true comparison (with respect to fronts), but since the same benchmark problems are attempted and the reference fronts are constructed using the same procedure, a reasonable approximate direct comparison can be done based on the graphical comparison of the obtained fronts, and the obtained values of Spread metric.

The Pareto fronts produced by MQHDE are shown in Figure 1. The MQHDE is found to be competitive in accomplishing the two conflicting goals of Multi-objective optimization problems. And also the range of the extreme points found by MQHDE and their values are better than NSGA II and MOCell in all problem instances[3]. From the plots, it is apparent that MQHDE is able to obtain the Pareto fronts with better spread and Hypervolume.

Table 1 reports the values of the Spread indicator. The table includes the values computed for Spread for NSGA II and MOCell after 10000 function evaluations [3] and MQHDE after 10000 and 5000 function evaluations. From the reported results it is evident that in the first test problem set concerned with 'scaling', MQHDE outperformed NSGA II and MOCell with significant differences after 5000 function evaluations itself. And also it is apparent that the algorithm maintained excellent diversity in the solutions even in typical cases. The performance of MQHDE is exceptionally good in most of the cases as is evident from the results shown in Table 1.



**Fig. 1** Pareto fronts obtained by MQHDE after 10000 function evaluations

The values of Hypervolume indicator are shown in Table 2. In all the problem instances MQHDE performed well and achieved better convergence and good diversity in the obtained Pareto-optimal front solutions. The difference in the values of Hypervolume obtained by MQHDE after 10000 and 5000 function evaluations is insignificant and it also shows that it is able to achieve superior results in half the number of evaluations being expended by other methods.

**Table 1** Results of Spread( $\Delta$ ) indicator\*

Customers - Requirements	NSGA II (10000) $\bar{x}_\sigma$	MOCcell (10000) $\bar{x}_\sigma$	MQHDE (10000) $\bar{x}_\sigma$	MQHDE (5000) $\bar{x}_\sigma$
15 – 40	4.93e-1 $\pm$ 4.2e-2	3.76e-1 $\pm$ 3.3e-2	3.29e-1 $\pm$ 2.6e-2	4.08e-1 $\pm$ 4.2e-2
50 – 80	5.00e-1 $\pm$ 3.5e-2	4.10e-1 $\pm$ 3.9e-2	2.96e-1 $\pm$ 3.0e-2	3.36e-1 $\pm$ 3.0e-2
100 – 140	5.51e-1 $\pm$ 3.7e-2	4.85e-1 $\pm$ 3.8e-2	3.12e-1 $\pm$ 2.5e-2	3.53e-1 $\pm$ 2.9e-2
100 – 20	7.98e-1 $\pm$ 9.1e-3	6.16e-1 $\pm$ 4.9e-3	6.53e-1 $\pm$ 3.9e-3	6.51e-1 $\pm$ 7.9e-3
100 – 25	5.79e-1 $\pm$ 2.4e-2	5.43e-1 $\pm$ 2.6e-2	6.00e-1 $\pm$ 6.4e-3	5.98e-1 $\pm$ 1.6e-2
20 – 200	6.06e-1 $\pm$ 3.6e-2	5.60e-1 $\pm$ 4.7e-2	3.14e-1 $\pm$ 2.9e-2	3.65e-1 $\pm$ 3.1e-2

**Table 2** Results of Hypervolume(HV) indicator\*

Customers - Requirements	NSGA II (10000) $\bar{x}_\sigma$	MOCcell (10000) $\bar{x}_\sigma$	MQHDE (10000) $\bar{x}_\sigma$	MQHDE (5000) $\bar{x}_\sigma$
15 – 40	6.63e-1 $\pm$ 1.6e-3	6.63e-1 $\pm$ 1.2e-3	6.66e-1 $\pm$ 5.4e-3	6.53e-1 $\pm$ 6.5e-3
50 – 80	5.88e-1 $\pm$ 4.6e-3	5.85e-1 $\pm$ 4.9e-3	6.56e-1 $\pm$ 9.8e-3	6.55e-1 $\pm$ 1.1e-2
100 – 140	5.28e-1 $\pm$ 6.4e-3	5.20e-1 $\pm$ 5.7e-3	6.68e-1 $\pm$ 1.1e-2	6.64e-1 $\pm$ 1.2e-2
100 – 20	6.13e-1 $\pm$ 2.1e-4	6.13e-1 $\pm$ 3.0e-4	6.52e-1 $\pm$ 7.0e-6	6.52e-1 $\pm$ 7.8e-4
100 – 25	6.35e-1 $\pm$ 6.3e-4	6.35e-1 $\pm$ 5.7e-4	6.53e-1 $\pm$ 2.7e-4	6.53e-1 $\pm$ 9.2e-4
20 – 200	5.26e-1 $\pm$ 7.6e-3	5.17e-1 $\pm$ 7.5e-3	6.76e-1 $\pm$ 1.1e-2	6.75e-1 $\pm$ 1.3e-2

\* In the above tables, the values preceding the  $\pm$  symbol are the mean ( $\bar{x}$ ) values and succeeding the  $\pm$  symbol in subscripts are the values of standard deviation ( $\sigma$ ).

## 7 Conclusions

This paper presents a new Multi-objective Quantum-inspired Hybrid Differential Evolution (MQHDE) Algorithm for the solution of MONRP which is an important problem in Search-Based Software Engineering and has wide applicability in software companies and manufacturing companies. MQHDE combines the strengths of Quantum Computing, Differential Evolution and Genetic Algorithms to achieve the right balance between exploration and exploitation of the search space. MQHDE delivers consistent performance in terms of convergence to the optimal front, maintaining good spread among the Pareto-optimal solutions, and fast convergence. The efficacy of MQHDE in the solution of Multi-Objective Next release problem is evaluated on six benchmark problems and the performance is compared against other state-of-the-art Multi-objective Evolutionary algorithms. The comparison is based on the obtained Pareto fronts, range of extreme points and the performance indicators: Spread ( $\Delta$ ) and Hypervolume (HV). The results indicate the superior performance of MQHDE over NSGA II, MOCcell and other methods, and also prove the effectiveness and efficiency of MQHDE in the solution of MONRP.



## References

1. Bagnall, A., Rayward-Smith, V., Whitley, I.: The next release problem. *Information and software technology* 43(14), 883–890 (2001)
2. Zhang, Y., Harman, M., Mansouri, A.S.: The Multi-Objective Next Release Problem. In: *GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pp. 1129–1136. ACM, New York (2007)
3. Durillo, J.J., Zhang, Y., Alba, E., Nebro, A.J.: A study of the multi-objective next release problem. In: *SSBSE 2009: Proceedings of the 2009 1st International Symposium on Search Based Software Engineering*, pp. 49–58. IEEE Computer Society, Washington (2009)
4. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, Chichester (2001)
5. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for solving Multi-Objective Problems*. Springer (2007)
6. Price, K., Storn, R.: *Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report, International Computer Science Institute, Berkeley (1995)
7. Han, K.H., Kim, J.H.: Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. *IEEE Transactions on Evolutionary Computation* 6(6), 580–593 (2002)
8. Defoin, M., Schliebs, S., Kasabov, N.: A Versatile Quantum-inspired Evolutionary Algorithm. In: *The IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 423–430 (2007)
9. Su, H., Yang, Y.: Quantum-inspired differential evolution for binary optimization. In: *The 4th International Conference on Natural Computation*, pp. 341–346 (2008)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
11. Zitzler, E., Thiele, L.: Multiobjective Evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4), 257–271 (1999)

# Evolving a Pareto Front for an Optimal Bi-objective Robust Interception Problem with Imperfect Information

Gideon Avigad, Erella Eisenstadt, and Valery Y. Glizer

**Abstract.** In this paper, a multi-objective optimal interception problem with imperfect information is solved by using a Multi-Objective Evolutionary Algorithm (MOEA). The traditional setting of the interception problem is aimed either at minimizing a miss distance for a given interception duration or at minimizing an interception time for a given miss distance. In contrast with such a setting, here the problem is posed as a simultaneous search for both objectives. Moreover, it is assumed that the interceptor has imperfect information on the target. This problem can be considered as a game between the interceptor, who is aiming at a minimum final distance between himself and the target at a minimal final time, and an artificial opponent aiming at maximizing these values. The artificial opponent represents the effect of the interceptor's imperfect information (measurement inaccuracies) on the success of the interception. Both players utilize neural net controllers that evolve during the evolutionary optimization. This study is the first attempt to utilize evolutionary multi-objective optimization for solving multi-objective differential games, and as far as our review went, the first attempt to solve multi-objective differential games in general.

## 1 Introduction and Problem Formulation

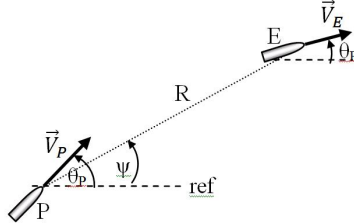
Consider a planar engagement between two objects moving at constant speed, an interceptor (a pursuer) and a target (an evader). The geometry of this engagement is shown in Fig. 1. This geometry defines the variables of the engagement. In these variables, the dynamics of the engagement is described by the following set of non-linear differential equations.

---

Gideon Avigad · Erella Eisenstadt · Valery Y. Glizer  
Department of Mechanical Engineering, Ort Braude College, Karmiel, Israel  
e-mail: [gideon,erella}@braude.ac.il](mailto:{gideon,erella}@braude.ac.il)

$$\dot{R} = V_e \cos(\theta_e - \psi) - V_p \cos(\theta_p - \psi), \quad (1)$$

$$\dot{\psi} = [V_e \sin(\theta_e - \psi) - V_p \sin(\theta_p - \psi)]/R. \quad (2)$$



**Fig. 1** The pursuer and the evader

The engagement (pursuit-evasion) starts at  $t = 0$  with the initial conditions:

$$R(0) = R_0, \quad \psi(0) = \psi_0. \quad (3)$$

The directions of the velocity vectors of the objects are governed by the dynamic controllers:

$$\dot{\theta}_p = \Omega_p u_p, \quad \theta_p(0) = \theta_{p0}, \quad (4)$$

$$\dot{\theta}_e = \Omega_e u_e, \quad \theta_e(0) = \theta_{e0}, \quad (5)$$

where

$$\Omega_p = \frac{V_p}{r_p}, \quad \Omega_e = \frac{V_e}{r_e} \quad (6)$$

are the maximal admissible turning rates of the pursuer and the evader, respectively;  $r_p$  and  $r_e$  are the smallest turning radii of the objects; and  $u_p(t)$  and  $u_e(t)$  are non-dimensional controls of the pursuer and the evader, respectively, satisfying the constraints:

$$|u_p(t)| \leq 1, \quad t \geq 0, \quad (7)$$

$$|u_e(t)| \leq 1, \quad t \geq 0, \quad (8)$$

The commonly treated differential game, associated with the dynamics (1)-(6) and the control constraints (7)-(8) (see e.g. [1] and references therein), is formulated as:

$$\min_{u_p} \max_{u_e}(t_f) \text{ subject to a given } R(t_f) = D_f > 0 \quad (9)$$

$$\text{or as } \min_{u_p} \max_{u_e}(R(t_f)) \text{ subject to a given } t = t_f > 0 \quad (10)$$

Clearly, this setting is not a multi-objective setting. The question is: what can be gained from setting the problem as a multi-objective differential game (MODG) such as (in most cases and  $t_f$  are contradicting):

$$\min_{u_p} \max_{u_e}(R(t_f), t_f) \quad (11)$$

$$\text{subject to: } 0 < R(t_f) \leq D_{is}, \quad 0 < t_f \leq T, \quad (12)$$

where  $D_{is}$  and  $T$  are boundaries that designate a region of interest. The foremost answer to the above question is itself a question rather than an answer: what is the decision-maker interested in. If he/she is interested in seeing tradeoffs, the MODG setting is preferable. Tradeoffs are of interest when the decision-maker wants some flexibility in choosing the preferable solution. Tradeoffs may allow the decision-maker an opportunity to consider whether to launch its champion pursuer with the most modern measuring system and maneuverability or to save money and launch an inferior pursuer that will still be accurate enough and, moreover, might intercept the target quicker. Suppose now that the decision-maker is interested in observing the Pareto front. Clearly, then, the choice of evolutionary algorithms for searching for the optimal solutions is a sensible choice. So the next question might be: why not take a sequential single-objective approach, that is, for each interception distance, minimize the interception time, or vice versa. There are two answers to this: a) using a multi-start single-objective approach for finding the Pareto front can result in a higher computational complexity than if a simultaneous multi-objective approach is considered (see e.g. [2]), and b) generally there is no guarantee that a finite final time exists for any given final distance.

## 2 Background

Most existing studies that deal with discrete multi-objective games (MOGs) are associated with definitions of the equilibrium. The predominant definition of this equilibrium, is the Pareto-Nash equilibrium, see e.g., [3]. The notion of the Pareto-Nash equilibrium is based upon the concept of cooperative games. According to Pareto-Nash, sub-players in the same coalitions should optimize their vector functions on a set of strategies. On the other hand, this notion also takes into account the concept of non-cooperative games, because coalitions interact among themselves on the set of situations and are interested in preserving Nash equilibrium between coalitions. An extension of the work of [3] can be found in [4]. Linear programming is commonly used for solving MOGs, e.g., [5], where a multi-objective zero-sum game is solved. A non-zero-sum version of a MOG was solved by [6]. Detailed algorithms

for finding strategies related to the Pareto-Nash equilibrium can be found in [3], [4]. A detailed mathematical step-by-step solution of a MOG using the Kuhn-Tucker equation can be found in [7].

In all of these algorithms, weights are altered in order to search for one equilibrium solution at a time. This means that sequential executions of the algorithms are needed to reveal more equilibrium points. Artificial intelligence-based approaches have been applied to MOGs within the framework of “fuzzy MOGs”. In such studies (e.g., [8]) the objectives are aggregated to a surrogate objective in which the weights (describing the players’ preferences towards the objectives) are modeled through fuzzy functions. Recently, Avigad et al. [9] used the worst case evolutionary algorithm suggested by [10] to evolve a set of optimal strategies for a discrete MOG. In the worst case analysis, each solution is associated with a set of scenarios. These scenarios are evolved in an embedded MOEA to find the worst scenario, which in a multi-objective problem (MOP) setting may be a set of worst scenarios. These worst scenarios represent the solution in an outer set-based evolutionary algorithm, in order to find the optimal solutions.

In contrast to discrete games, in differential games the players’ moves are found simultaneously by solving the related model, which is described by differential equations. To the best of our knowledge, multi-objective differential games have not been considered in the literature.

### 3 Methodology

This section is organized as follows: Section 3.1 deals with the evolution of an optimal control for the pursuer so as to simultaneously minimize the interception time and distance. In this section, no game is yet played, and a Pareto front is evolved as a reference for the game that follows. In Section 3.2, the game is posed and formulated. Moreover, the path for finding the pursuer optimal control is set and the strived-for result is formulated. Section 3.3 introduces the multi-objective evolutionary algorithm that enhances the search for the pursuer optimal control in the presence of uncertainties. Section 3.4 demonstrates the work of the algorithm on the game at hand.

#### 3.1 *Pursuer’s Control in a No-Game Situation*

As a first step, a MOP rather than a single-objective problem (SOP) is solved in order to evolve a front that will serve as a reference for the game to follow (see Section 3.2). The problem is degraded to a MOP rather than a MOG by considering  $\theta_e(t)$  as a predefined function. This means there are no interactive controls or, in other words, there is no game. The control problem is set as a MOP, which is defined by

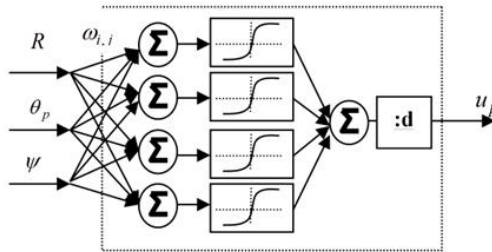
the equations of dynamics (1)-(2), the initial conditions (3), the controller (4), the control constraint (7) and the multi-objective performance index:

$$\min_{u_p} (R(t_f), t_f) \tag{13}$$

subject to the inequalities

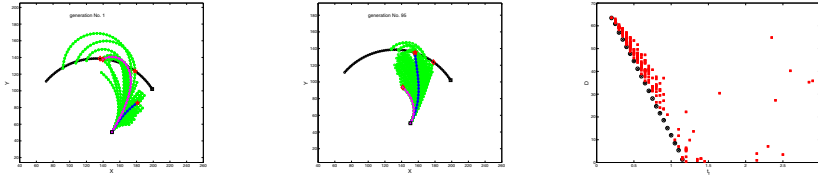
$$0 < R(t_f) \leq D_{is}, \quad 0 < t_f \leq T, \tag{14}$$

For the purpose of finding optimal controls, an artificial neural network (ANN) controller is considered. The controller consists of  $d$  nodes with sigmoid functions as activation functions. This means that the design of the controller amounts to selecting a set of weights  $\omega_{i,j}$  for the  $i$ -th input and  $j$ -th node. As is common when designing a feedback pursuer's control,  $R$ ,  $\theta_p$  and  $\psi$  are the control inputs and therefore  $u_p = [\omega_{1,1}, \dots, \omega_{i,j}, \dots, \omega_{d,3}]^T$ . The weights are evolved by using an evolutionary multi-objective optimization algorithm (here NSGA-II, [11] is utilized). The choice on NSGA-II is arbitrary and other MOEAs might be used. The small number of objectives and the simplicity of the problem make this algorithm good enough.



**Fig. 2** The NN pursuer's controller. Division by 'd' ensures that the controller's output is bounded between -1 and 1.

The left panel of Figure 3 depicts the evader's trajectory, designated by the black curve, which is constructed from circles. Also depicted are fifty pursuers' trajectories, obtained by solving equations (1)-(2), while using the initial generation's population that decodes ANN controls, each with four nodes ( $d=4$ ). In this figure, the two trajectories are marked in purple for the trajectory with the least interception distance and in blue for the one with the minimal interception time. The diamonds and stars in this figure respectively designate the location of the pursuer and of the evader for the minimal interception time and distance. The middle panel of Figure 3 depicts the same, but for the final population.

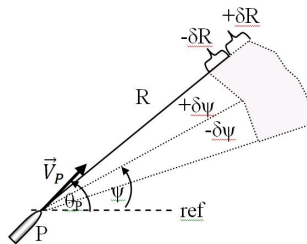


**Fig. 3** Left panel: The initial population's trajectories are depicted by green circles, the evader's trajectory by black circles, the pursuer and evader starting points by black squares, the interception minimal distance and minimal time trajectories by purple and blue curves, respectively. Middle panel: The Pareto set trajectories are depicted showing diverse trajectories, including minimal interception distance and time (blue and purple trajectories respectively). Right panel: The initial population performances (red squares) and the Pareto front (circled crosses).

The middle panel highlights the aquacade interception. The locations of the pursuer and of the evader are identical at the interception point (where the two stars overlap). It is also interesting to note that in order to attain minimal interception time, the evolved controller causes the pursuer to follow the evader, but from a distance (depicted by the purple trajectory). The right panel of Figure 3 depicts the performances of the initial population designated by red squares and the evolved Pareto front (generation 95) designated by circled crosses. The evolved front will serve as a reference for the simulations carried out in the following section.

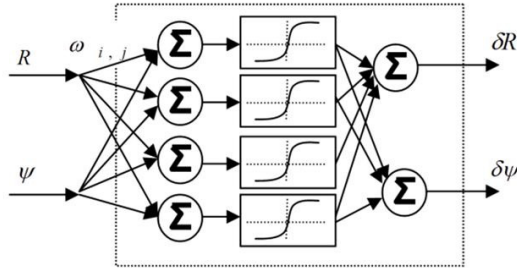
### 3.2 Pursuit with Uncertainties: The Game Situation

In Section 3.1 no game was played because there was just one player with no active opponent. To upgrade the problem, we consider an uncertainty, in this case the uncertainty associated with measurement inaccuracy. This means that the pursuer's measurements of  $R$  and  $\psi$  are uncertain such that:  $R = R \pm \delta R$  and  $\psi = \psi \pm \delta \psi$ . The new situation is depicted in Figure 4.



**Fig. 4** The different parameters and the related uncertainties

Similar to the pursuer’s controller, an artificial controller is constructed for the uncertainty. Its inputs are the uncertain parameters  $R$  and  $\psi$ . The outputs of the controller are  $\delta R$  and  $\delta\psi$ . These values are added to  $R$  and  $\psi$ . Then, the obtained results are used as inputs to the pursuer’s control. Figure 5 depicts this neural net controller.



**Fig. 5** The artificial opponent’s NN controller, evolved in order to produce the worst measures of  $R$  and  $\psi$  at every time step

The game is defined as a game between a pursuer that aims at minimizing the interception distance and time by manipulating  $u_p$ , while the artificial controller aims at maximizing these objectives by manipulating the values of  $\delta R$  and  $\delta\psi$ . The purpose of such a game is to find the best performances of the pursuer control under the worst conditions. Posing such a problem may supply results that ensure high reliability (by using worst case analysis).

Thus, the above mentioned game consists of the equations of dynamics (1)-(3), the initial conditions (3), the dynamic pursuer’s controller (4), the pursuer’s control constraint (7) and the performance index:

$$\min_{u_p} \max_{(\delta R, \delta\psi)} (R(t_f), t_f) \text{ subject to the inequalities (12).} \tag{15}$$

In this game, the pair  $(\delta R, \delta\psi)$  is a control of the pursuer’s artificial opponent, playing on the evader’s side. This control satisfies the inequalities:

$$-\Delta R \leq \delta R \leq \Delta R, -\Delta \psi \leq \delta \psi \leq \Delta \psi, \tag{16}$$

where  $\Delta R, \Delta\psi$  are positive numbers known to the pursuer. Due to these inequalities, for each  $u_p$ , there may be a cluster of scenarios in objective space associated with different pairs  $c = (\delta R, \delta\psi)$ . The pursuer’s controller and the artificial controller are ANN controllers with  $d$  and  $e$  nodes respectively, such that:  $u_p^{(s)} = [\omega_{1,1}^{p(s)}, \dots, \omega_{i,j}^{p(s)}, \dots, \omega_{d,3}^{p(s)}]^T, c^{(m)} = [\omega_{1,1}^{c(m)}, \dots, \omega_{i,j}^{c(m)}, \dots, \omega_{e,2}^{c(m)}]^T$ . Let  $U$  and  $C$  be the sets of all possible  $S$  and  $M$  controllers for the pursuer and the artificial opponent, respectively, such that:



$U = \{u_p^{(1)}, \dots, u_p^{(s)}, \dots, u_p^{(S)}\}$ ,  $C = \{c^{(1)}, \dots, c^{(m)}, \dots, c^{(M)}\}$ , where  $u_p^{(s)} \in \Omega \subseteq R^S$ ,  $c^{(m)} \in \Gamma \subseteq R^M$ . A game outcome is a result of the opponent's choice of controls  $u_p^s$  and  $c^{(m)}$ . Thus, the above formulated game between the pursuer and the artificial opponent is reduced to a game between two players, using the controls  $u_p^{(s)}$  and  $c^{(m)}$ . This pair of controls is  $g_{s,m} = (u_p^{(s)}, c^{(m)}) \in \Phi = \Omega \times \Gamma$ . The performance index can be rewritten as:

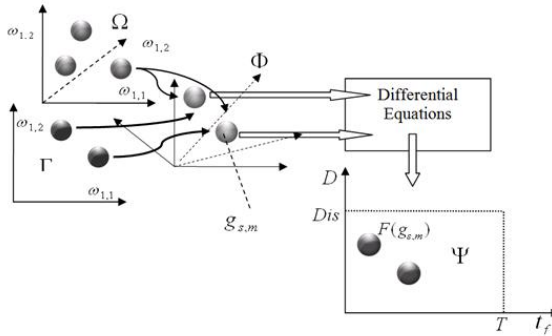
$$\max_{u_p^{(s)}} \min_{c^{(m)}} F(g_{s,m}), \quad (17)$$

$$F(g_{s,m}) = (R(t_{f,s,m}), t_{f,s,m}) \in \Psi \subseteq R^2 \quad (18)$$

subject to the inequalities

$$0 < R(t_{f,s,m}) \leq D_{is}, 0 < t_{f,s,m} \leq T, \quad (19)$$

where  $\Psi$  is the set of all possible game outcomes,  $R(t_{f,s,m})$  and  $t_{f,s,m}$  are the distance and the time, generated by the s-th pursuer control and the m-th artificial opponent control, respectively. The value of the function F is in fact the result of solving the differential equations of the game that utilize the sets of weights of the two controls. Figure 6 depicts the different notions described and formulated above. It highlights the different spaces and the way in which the game is constructed.



**Fig. 6** The different notions put forward hereby

In the presence of uncertainties, solution reliability is commonly a major issue. In this paper we consider worst-case reliability. This means that we aim for 100 percent success. The demand for Reliability=1 may be translated to a test for feasibility. The feasible pursuer's control is a control for which the following is fulfilled:  $u_p^s \in U$ , is feasible  $\Leftrightarrow \neg \exists c^{(m)} \in C : \{R(t_{f,s,m}) > D_{is} \vee t_{f,s,m} > T\}$ .

In a MOP, the worst case might be (in the presence of contradicting objectives) a set of worst scenarios within the objective space. For each of the pursuer

controls (say the  $s$ -th strategy),  $M$  controls are possible for the artificial entity. Therefore, there may be  $M$  games. All such possible games form a set:  $G_{u_p^{(s)}} \subseteq \Phi = [g_{s,1}, \dots, g_{s,m}, \dots, g_{s,M}]^T$ . The  $m$ -th artificial opponent aims at maximizing the distance and time attained in a game with the  $s$ -th pursuer, that is:

$$\max_{c^{(m)}} F(g_{s,m}), \quad m = 1, \dots, M \quad (20)$$

Clearly, the objectives of the game (minimizing  $R(tf)$  and  $tf$ ) contradict one another. Hence, there is a set of artificial opponent controls that may serve as solutions to this problem. These will form a Pareto set of artificial opponent controls playing against the  $s$ -th pursuer controller:

$$C_{u_p^{(s)}}^* : \{c^{(m)} \in C \mid \neg \exists c^{(m')} : F(g_{s,m'}) \succ F(g_{s,m})\}, \quad \forall m, m' = 1, \dots, M \quad (21)$$

Note that here  $\succ$  is used instead of  $\prec$ . Mapping these optimal artificial opponent controllers to the objective space forms a Pareto front  $F_{u_p^{(s)}}^*$ , where  $F_{u_p^{(s)}}^* : \{F(g_{s,m}) \in \Psi \mid c^{(m)} \in C_{u_p^{(s)}}^*\}$  is associated with the  $s$ -th pursuer's controller.

Now we can search for the optimal pursuer control, subject to:

$$\min_{u_p^s \in U} F_{u_p^s}^*, \quad s = 1, \dots, S \quad (22)$$

This means that the best pursuer control is searched for, while considering the best (most destructive) control used by the artificial opponent. This optimization results in a set of optimal pursuer controls:

$$U^* := \{u_p^s \in U \mid \forall c^{(m)} \in C \neg \exists u_p^{s'} \in U : F(g_{s',m}) \prec F(g_{s,m})\} \quad (23)$$

for all  $m$  and for all  $s \neq s'$

Here we take the high reliability for interception one step further by representing each cluster by its related ideal point (the utopia point related to the extremes of the performances in objective space). That is, we represent each control by a point in the objective space such that:

$$\vec{F}(u_p^{(s)}) = [\max(R(t_{f_{s,m}})), \max(t_{f_{s,m}})] \quad (24)$$

Figure 7 is used to elucidate the notions introduced by equations 22-24. The figure depicts four clusters in the objective space, designated by different colors. Each cluster represents the performances in objective space of games played between one opponent's controller and all possible controls for the artificial opponent. In fact, the  $s$ -th cluster involves  $F(g_{s,m})$  for all  $m=1, \dots, M$ .

The Pareto front  $F_{u_p^{(s)}}^*$  of each cluster is designated by enlarged symbols. The ideal points for these fronts are designated by pluses. Representing each cluster by the ideal point highlights the following points:

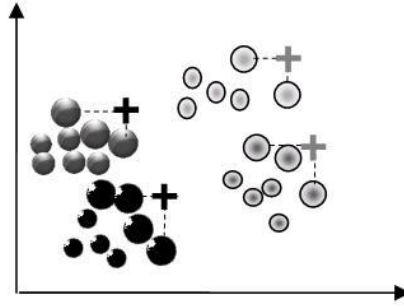


Fig. 7 Four pursuers' related scenarios, fronts and ideal points

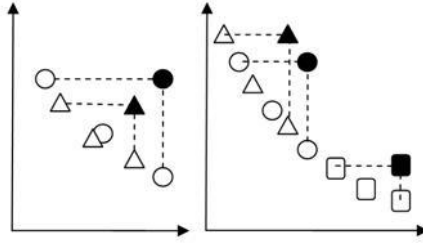
- a) Clearly, if any of the scenarios of a solution violates the constraints, so does the ideal point
- b) A solution with more scenarios spread along the front is worse (from the pursuer's viewpoint) than a non-spread scenario. This is because the evader, by having more spread scenarios, may possess higher flexibility in determining the tradeoff point between distance and time. Choosing the ideal point as a representative point confirms this observation. The left panel of Figure 8 depicts the  $F_{u_p^0}^*$  of two candidate pursuer controls differentiated one from the other by circles and triangles. Also shown are their related ideal points,  $\vec{F}(u_p^0)$ , designated by bold symbols. Note that the ideal point related to spread scenarios dominates the ideal point of less spread scenarios when maximization is considered (worse if minimization is considered).
- c) If the worst fronts associated with several pursuer controls are crowded, so are their related ideal points. The right panel of Figure 8 depicts the  $F_{u_p^0}^*$  of three candidate pursuer controls differentiated one from the other by circles and triangles. Also shown are their related ideal points,  $\vec{F}(u_p^0)$ , designated by bold symbols. Clearly the triangle- and circle-related fronts are crowded, whereas the square-related front is rather isolated. This relative situation does not change when the related ideal points are considered, as may be observed by depicting these points, which are also depicted in the figure.

Now that each  $u_p^{(s)}$  is represented by its  $\vec{F}(u_p^{(s)})$ , the Equation (23) can be rewritten as:

$$U^* := \{u_p^s \in U \mid \forall c^{(m)} \in C \neg \exists u_p^{s'} \in U : \vec{F}(u_p^{s'}) \prec \vec{F}(u_p^s)\} \quad (25)$$

*for all m and for all s ≠ s'*

If there were only four possible pursuer controls, then the controls represented by the black pluses in Figure 7 would be the optimal controls and would belong to  $U^*$ .



**Fig. 8** Left panel: This panel highlights the fact that a spread front is associated with an ideal point that dominates the ideal point of a less spread front. Therefore, from the point of view of the pursuer (that is, minimizing the objectives), the less spread front is better. Right panel: Crowded fronts involve crowded ideal points.

### 3.3 The MOE Algorithm

The evolutionary algorithm for searching for the optimal controls is outlined below. It consists of inner and outer loops. In the inner loop, an embedded algorithm searches for the worst set for each of the candidate controls by running a MOEA (e.g., NSGA-II). It is noted again that the embedded MOEA optimizes the solutions for the reversed problem, and therefore the algorithmic conditions should be adapted accordingly. The outer loop evolves the pursuer control based on the ideal point, which is obtained from the inner loop.

#### A. The Algorithm

- a. Initialize a population  $P_t$  of size  $N = |P_t|$  of pursuer controls up (weights of the NN controllers). Also, set  $Q_t = P_t$ .
- b. Combine parent and offspring populations and create  $R_t = P_t \cup Q_t$

The Embedded Algorithm (inner loop)

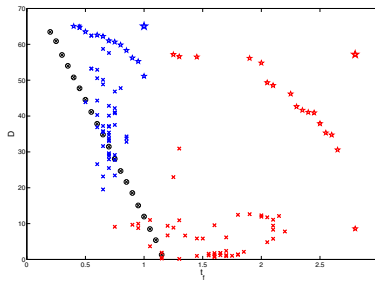
- c. For each individual of  $R_t$ :
  - c.1 Initialize a population  $G_t$  of size  $ng = |G_t|$  for the auxiliary opponent's controllers  $C^{(m)}$  (weights of NN controllers)
  - c.2 Run NSGA-II on the reversed optimization problem to find, for each  $u_p$  of  $R_t$ , the corresponding  $C_{u_p}^{*(s)}$  using  $F(g_{s,m})$ .
  - c.3 For each  $u_p$  of  $R_t$  use  $F_{u_p}^{*(s)}$  to assign  $F(u_p^{(s)})$ .
- d. Initialize a new parent population  $P_{t+1} = \emptyset$
- e. Assign a non-dominance level and crowding value (NSGA-II) to all individuals of  $R_t$  by using  $F(u_p^{(s)})$ .
- f. Fill in  $P_{t+1}$  according to their level of non-dominance and crowding measure.

- g. Create an offspring population  $Q_{t+1}^*$  from  $P_{t+1}$  by tournament selection.
- h. Perform Crossover on  $Q_{t+1}^*$  to obtain  $Q_{t+1}^{**}$ .
- i. Perform mutation to obtain  $Q_{t+1}$ .
- j. If the last generation has not been arrived at, go-to 'b'.

The embedded MOEA is computationally expensive. The need for a complete evolutionary run for each individual in a population is a major drawback of the approach. Without considering the complexity of the non-embedded algorithm, the embedded algorithm complexity is profoundly higher than that of a common MOEA. The complexity of NSGA-II is  $O(GKn^2)$ , where  $G$  is the number of generations,  $K$  is the number of objectives and  $n$  is the population size. The  $n^2$  term is due to fitness assignment. Here, for each solution the complexity is:  $O(EK(ng)^2)$  where  $E$  is the number of generations of the embedded algorithm. Therefore, the overall complexity becomes  $O(GnEK(ng)^2)$ . However, the approach is tractable and, moreover, there are several ways to reduce this complexity. Such approaches are described in Avigad and Branke 2008. Alternative methods can be found in Branke 2008.

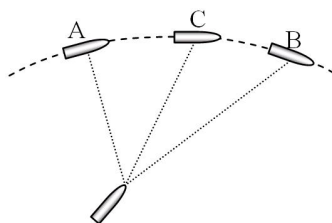
### 3.4 Simulations for the Game at Hand

Considering the problem at hand with 3% uncertainty at distance  $R$  and angle  $\psi$ , Figure 9 depicts the Pareto front of Figure 3 as well as the clusters of performances related to two initial population pursuer controls associated with different possible (uncertain) games. The clusters are designated by the red and blue crosses. The inner loop evolves the Pareto fronts of these clusters. These are depicted in the figure by the small blue and red stars. The ideal points for these clusters, which are computed at the end of the inner loop, are depicted in the figure by the large stars. These represent the candidate controls in the outer loop evolution.



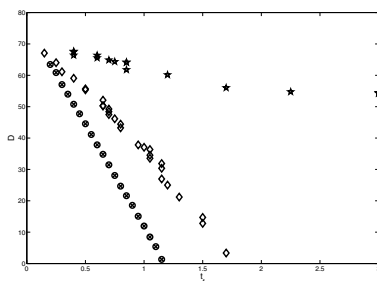
**Fig. 9** The no-uncertainty Pareto front (encircled crosses), two clusters of performances belonging to an initial population (red and blue crosses), their evolved fronts (small stars) and their related ideal points (large stars)

The performances in the initial cluster, which appear to be more optimal than the Pareto front, must be considered here. To explain this seemingly problematic issue, consider Figure 10. The figure depicts the pursuer, the evader and three possible measurements of  $R$  and  $\psi$ . The measurements are designated by dashed lines. The three measurements include two false measures (A and B) and a true measure (no uncertainty) - measure C. Measure C would probably yield a better control action (more explosive turn of the pursuer), as it “predicts” the next evader movement. Such a prediction (falsely achieved) may lead to better interception. In the first generation’s cluster, such measures are present, yielding the points that dominate the no-uncertainty Pareto front. However, because accuracy is not controlled, another set of measures (e.g., involved with line A) may cause the pursuer to turn in the wrong direction. The latter measures are those that are most distracting and are “preferred” by the artificial control.



**Fig. 10** Two false measures (A and B) and an accurate measure, C. Measures A and B may lead to excellent moves based on prediction (falsely attained) B, or to terrible controls (in the case of A).

Figure 11 depicts the no-uncertainty in a measures-related front (circled crosses), as well as the fronts involved with games with 1% and 3% inaccuracy in measurements, designated by diamonds and stars, respectively.



**Fig. 11** Three Pareto fronts: No mistakes in measurements (circled crosses), 1% inaccuracy and 3% inaccuracy in measurements (diamonds and stars respectively)

The results clearly depict the problematic influence of inaccuracy in measuring  $R$  and  $\psi$ . Usually the uncertainty associated with measurements may be accurately assessed. Based on this knowledge, decision-makers may choose a controller that will ensure interception even though uncertainty exists.

## 4 Summary and Conclusions

This paper describes a novel attempt to motivate the need for and to optimize controllers of two opponents playing a multi-objective differential game. In contrast to the common interception problem where either miss-distance or time-to-interception is minimized, here these two objectives are simultaneously minimized. Optimizing these objectives simultaneously may provide a decision-maker enhanced flexibility in choosing the controller. It is shown that for each possible controller chosen by one opponent, there may be a set of optimal controllers for the other opponent. Therefore, in order to find optimal controllers for the first opponent, a set-based search has to be conducted. In this paper the problem and its solution are defined, and an evolutionary multi-objective optimization algorithm is proposed. The working of the algorithm is demonstrated on an interception game, where one opponent has to optimize controllers for a pursuer, while the other opponent involves anti-optimization of the same objectives by maximizing the effect of uncertainty on the interception. It is shown that using the suggested approach may allow decision-makers to choose controllers for pursuers while taking into account the worst-case influence of uncertainty.

Solving differential games is known as a difficult task. The mathematics involved hindered treating these problems as MODGs. We believe that in this work we have paved the way for treating such problems by utilizing evolutionary algorithms. Moreover, it seems that considering MODGs will pave the way for new differential games and for suggesting solutions to yet unconsidered problems.

**Acknowledgements.** This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.

## References

1. Glizer, V.Y., Shinar, J.: On the Structure of a Class of Time Optimal Trajectories. *Optimal Control Applications & Methods* 14, 271–279 (1993)
2. Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., Deb, K.: Running Time Analysis of Multi-objective Evolutionary Algorithms on a Simple Discrete Optimization Problem. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) *PPSN 2002*. LNCS, vol. 2439, pp. 44–53. Springer, Heidelberg (2002)
3. Zelikovskiy, A., Dmitrii, L., Solomon, D.: Multiobjective games and determining Pareto-Nash equilibria. *Buletinul Academiei De Stiinte, A Republicii Moldova Matematica* (3), 115–122 (2005)

4. Somasundaram, K., Baras, J.S.: Pareto Nash Replies for Multi-Objective Games. The Institute for Systems Research ISR technical report 2009-4 (2009)
5. Zeleny, M.: Games with multiple payoffs. *International Journal of Game Theory* 4(4), 179–191 (1975)
6. Contini, B.M.: A decision model under uncertainty with multiple payoffs. In: Mensch, A. (ed.) *Theory of Games; Techniques and Applications*, pp. 50–63. American Elsevier Pub. Co. (1966)
7. Susan, X.L.: Interactive strategy sets in multiple payoff games. *Computers & Industrial Engineering* 37(3), 613–630 (1999)
8. Sakawa, M.: *Genetic Algorithms and Fuzzy Multiobjective Optimization*. Kluwer Academic Publishers, Boston (2002)
9. Avigad, G., Eisenstadt, E., Weiss, M.: Optimal Strategies for Multi Objective Games and Their Search by Evolutionary Multi Objective Optimization. In: *Proceedings of the Conference on Computational Intelligence in Games (CIG 2011)*, Seoul, South Korea, August 31-September 4 (2011) ISBN: 978-14577-0009-5
10. Avigad, G., Branke, J.: Embedded Evolutionary Multi-Objective Optimization for Worst Case Robustness. In: *2008 Genetic and Evolutionary Computation Conference (GECCO 2008)*, pp. 617–624. ACM Press, USA (2008) ISBN 978-1-60558-131-6
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6(2), 182–197 (2002)



# PSA – A New Scalable Space Partition Based Selection Algorithm for MOEAs

Shaul Salomon, Gideon Avigad, Alex Goldvard, and Oliver Schütze

**Abstract.** It has generally been acknowledged that both proximity to the Pareto front and a certain diversity along the front should be targeted when using evolutionary algorithms to evolve solutions to multi-objective optimization problems. Although many evolutionary algorithms are equipped with mechanisms to achieve both targets, most give priority to proximity over diversity. This priority is embedded within the algorithms through the selection of solutions to the elite population based on the concept of dominance. Although the current study does not change this embedded preference, it does utilize an improved diversity preservation mechanism that is based on a recently introduced partitioning algorithm for function selection. It is shown that this partitioning allows for the selection of a well-diversified set out of an arbitrary given set. Further, when embedded into an evolutionary search, this procedure significantly enhances the exploitation of diversity. The procedure is demonstrated on commonly used test cases for up to five objectives. The potential for further improving evolutionary algorithms through the use of the partitioning algorithm is highlighted.

## 1 Introduction

In many problem situations, several objectives must be optimized at the same time, leading to a *multi-objective optimization problem* (MOP). Mathematically, a MOP can be stated as follows:

$$\min_{x \in Q} \{F(x)\}, \tag{1}$$

---

Shaul Salomon · Gideon Avigad · Alex Goldvard  
ORT Braude College of Engineering, Karmiel, Israel  
e-mail: [shaulsal, gideon, goldvard}@braude.ac.il](mailto:{shaulsal, gideon, goldvard}@braude.ac.il)

Oliver Schütze  
Computer Science Department, CINVESTAV-IPN, Av. IPN 2508,  
Col. San Pedro Zacatenco, Mexico City, Mexico  
e-mail: [schuetze@cs.cinvestav.mx](mailto:schuetze@cs.cinvestav.mx)

where  $Q \subset \mathbb{R}^d$  is the domain,  $F$  is defined as the vector of the  $k$  objective functions  $F : Q \rightarrow \mathbb{R}^k$ ,  $F(x) = (f_1(x), \dots, f_k(x))$ , and each objective is defined as  $f_i : Q \rightarrow \mathbb{R}$ . The set of optimal solutions of a MOP, the *Pareto set*  $\mathcal{P}$ , typically forms a  $(k - 1)$ -dimensional object. The task of many set-oriented search procedures is to find a ‘suitable’ finite sized approximation of the *Pareto front*  $F(\mathcal{P})$  (i.e., the image of the Pareto set), since this front represents the set of optimal compromises measured in objective space, the most interesting view in many applications.

Of the set-oriented search procedures for the numerical treatment of MOPs, multi-objective evolutionary algorithms (MOEAs) are widely used due to their global and universal approach and their high robustness ([6], [5]). Most MOEAs simultaneously attempt to account both for the proximity of the approximation set to the Pareto front (also termed as *convergence* in the community) and for its diversity (also termed as *spread*) [3]. Domination is the predominant approach used to exploit proximity. Diversity is exploited by different approaches that can be classified into three main categories. The first treats diversity as a property of a set and evolves sets with good diversity. The diversity can be measured according to the accumulated distances between the members of the set (see [12], [13]), or indirectly by the hypervolume measure [20] or the averaged Hausdorff distance [14]. Algorithms in the second category treat diversity as a property of each individual according to the density of solutions surrounding it. Fitness sharing of NPGA [10], crowding distance of NSGA-II [7], the diversity metric based on entropy [15] and the density estimation technique used in SPEA2 [22] are examples of this category. Algorithms of the third category decompose the multi-objective problem into a number of single objective problems (scalarization). Each of these problems ideally aims for a different zone on the Pareto front such that the set of solutions to the auxiliary problems form a diverse set of optimal solutions. MOEA/D [19] is probably the most famous method within this category.

With respect to both goals (proximity and diversity), selection plays an important role in evolving individuals towards the Pareto front [1]. In order to select promising (i.e., less dominated and less crowded) individuals, proper selection criteria must be formulated. These criteria are commonly utilized in MOEAs in order to choose the elite population, thus allowing the best solutions in the current generation to be copied into the next generation. An elite population may be selected by pure truncation selection. In truncation selection with threshold  $\tau$ , the algorithm sorts all individuals based on their domination level and includes the first  $\tau$  individuals as the elite population. Truncation selection is exploited in many MOEAs, such as NSGA-II [7] and SPEA2 [22]. In this case, exploitation of proximity takes over exploitation of diversity. Some efforts to further promote population diversity have been made over the last decade, among them Balanced Truncation Selection (BTS) [3], epsilon Dominance Selection [11], Indicator-Based Selection, IBS [21].

In this paper, a new algorithm, PSA, is proposed to select a diverse subset from a given set of points. The proposed algorithm has several advantages that make it suitable for integration into MOEAs: 1) It has low computational complexity. 2) It can select a diverse subset, even if the original set is poorly distributed. 3) There are no limitations on the size of the selected subset.

The remainder of this paper is organized as follows. The proposed algorithm is described in Section 2. Section 3 describes a straightforward method for integrating the algorithm into NSGA-II in order to improve the diversity of its population. The proposed algorithm and the original NSGA-II are compared in Section 4. Finally, the paper ends with some conclusions in Section 5.

## 2 PSA – Part and Select Algorithm

This section proposes an algorithm that aims at selecting  $m$  well-spread points from a set of  $n$  candidate solutions. The proposed method is an adaption of the algorithm for the selection of functions (see [2]) to the context of multi-objective optimization. It selects a diverse subset from a set of objective vectors (e.g., the image of the current population of a MOEA). The procedure has two steps: First, the set is divided into subsets so that similar members are grouped in the same subset. Next, a diverse subset is formed by selecting one member from each generated subset.

### 2.1 Partitioning a Set

The core of the PSA is the algorithm of partitioning a given set of points in the objective space into smaller subsets. In order to partition a set into  $m$  subsets, PSA performs  $m - 1$  divisions of one single set into two subsets. At each step, the set with the greatest dissimilarity among its members is the one that is divided. This is repeated until the desired stopping criterion is met. The criterion can be either a predefined number of subsets (i.e., the value of  $m$ ) or a maximal dissimilarity among each of the subsets. The dissimilarity of a set  $A$  is defined by the measure  $\varnothing A$  as follows:

Let  $A := \{\mathbf{f}_1 = [f_{11}, \dots, f_{1k}], \dots, \mathbf{f}_n = [f_{n1}, \dots, f_{nk}]\} \subset \mathbb{R}^k$  (i.e.,  $n$  objective vectors  $\mathbf{f}_i = F(x_i)$  for points  $x_i \in Q$ ), and denote

$$a_j := \min_{i=1, \dots, n} f_{ij}, \quad b_j := \max_{i=1, \dots, n} f_{ij}, \quad \Delta_j := b_j - a_j, \quad j = 1, \dots, k \quad (2)$$

$$\varnothing A := \max_{j=1, \dots, k} \Delta_j \quad (3)$$

In fact,  $\varnothing A$  is the diameter of the set  $A$  in the Chebyshev metric. The size of  $\varnothing A$  is a measure of the dissimilarity among the members of  $A$ , with a large  $\varnothing A$  indicating a large dissimilarity among the members of  $A$ .

The pseudocode of PSA for a fixed value of  $m$  is shown in Algorithm 1. At every iteration the algorithm finds the subset with the largest diameter, and divide it into two subsets.

Figure 1 demonstrates the steps of the algorithm and highlights the results obtained by its use. Consider the set of 24 points in the bi-objective space depicted in the top left panel of Figure 1. Suppose that the purpose is to partition this set into  $m = 5$  subsets. The gray rectangle represents the region in the objective space that contains the solutions of the set. According to Eq. (3), the diameter of the given set

---

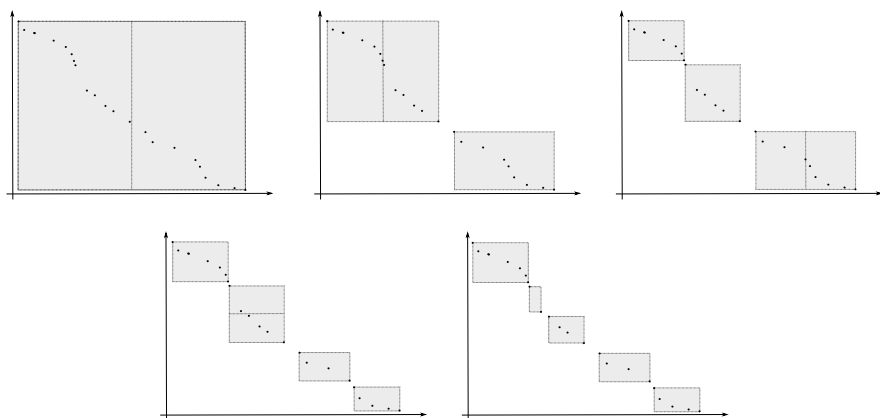
**Algorithm 1** Partitioning a set  $A$  into  $m$  subsets
 

---

- 1:  $A_1 \leftarrow A$
  - 2: Evaluate  $\varnothing A_1$  according to Eq. (3) and store  $\varnothing A_1$  in an archive.
  - 3:  $i \leftarrow 2$
  - 4: **while**  $i < m$  **do**
  - 5: Find  $A_j$  and coordinate  $p_j$  such that  $\varnothing A_j = \Delta_{p_j} = \max_{l=1, \dots, i-1} \varnothing A_l$
  - 6: Part  $A_j$  to subsets  $A_{j_1}, A_{j_2}$ :  
 $A_{j_1} \leftarrow \{\mathbf{f} = [f_1, \dots, f_{p_j}, \dots, f_k] \in A_j, f_{p_j} \leq a_{p_j} + \varnothing A_j / 2\}$   
 $A_{j_2} \leftarrow \{\mathbf{f} = [f_1, \dots, f_{p_j}, \dots, f_k] \in A_j, f_{p_j} > a_{p_j} + \varnothing A_j / 2\}$
  - 7: Evaluate  $\varnothing A_{j_1}$  and  $\varnothing A_{j_2}$  according to Eq. (3), and replace in the archive  $\varnothing A_j$  and  $p_j$  with the pairs  $\varnothing A_{j_1}, \varnothing A_{j_2}$  and  $p_{j_1}, p_{j_2}$  accordingly.
  - 8:  $S \leftarrow \{A_1, \dots, A_{j_1}, A_{j_2}, \dots, A_i\}$
  - 9:  $i \leftarrow i + 1$
  - 10: **end while**
- 

is the length of the horizontal side of the rectangle. Therefore, the first partition is made by vertical incision (indicated by the vertical line in the middle of the rectangle). The results of this partition are depicted in the top middle panel of Figure 1. The left subset in this panel has the greatest diameter (in horizontal direction). Therefore, the next partition is made on this subset by vertical incision. The results of this partition are depicted in the top right panel of Figure 1. The bottom two panels of Figure 1 depict the results of the next two iterations of Algorithm 1.

Note that the results of the partitioning are different than the results of using a common grid in the original space. With a common grid, an initial interval in every dimension is divided into equal sections, resulting in the division of the hyperbox into smaller hyperboxes of equal space. Since the original set  $A$  does not necessarily ‘cover’ the entire space, each hyperbox in the grid might or might not contain a member of  $A$ . Hence, there is no way to predict which resulting grid will have the desired number of occupied boxes. In addition, there are certain limitations on the number of hyperboxes in the grid. For example, in a two-dimensional grid it is possible to create  $m = \{1, 2, 4, 6, 9, 12, \dots\}$  boxes, while only a number of  $m = n^2$ , when  $n$  is a positive integer, will produce an even grid. With PSA, only the occupied space (marked as the gray rectangles in Figure 1) is considered. When a group  $A_i$  is divided into two subgroups  $A_{i_1}$  and  $A_{i_2}$ , the space considered from now on is given only by the two hyperboxes circumscribing  $A_{i_1}$  and  $A_{i_2}$ . The rest of the space in  $A_i$  is discarded. Every partition increases the number of subgroups by one, and therefore any desired number of subgroups can be created. Moreover, by dividing the subgroup with the largest dissimilarity, PSA ensures that the members of each group will be as similar as possible to each other.



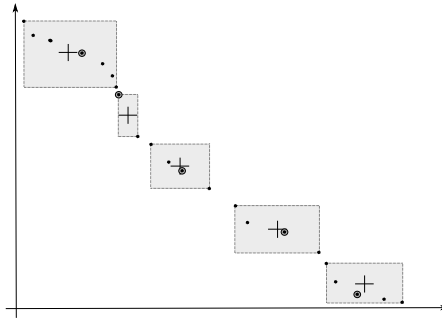
**Fig. 1** Partitioning of 24 elements in bi-objective space into  $m = 5$  subgroups (indicated by the gray boxes)

## 2.2 Selection of a Representative Subset

Once the set  $A$  has been divided into the  $m$  subsets  $A_1, \dots, A_m$ , the ‘most suitable’ element from each subset must then be chosen in order to obtain a subset  $A_{(r)}$  of  $A$  that contains  $m$  elements. This is of course problem dependent. Since this study aims for high diversity of the chosen elements, the following heuristic is suggested (denoted as center point selection): From each set  $A_i$  choose the member which is closest (in Euclidean metric) to the center of the hyperrectangle circumscribing  $A_i$ . If there exist more than one member closest to the center, one of them is chosen randomly.

Figure 2 illustrates this rule. The original set of 24 elements (compare to Figure 1) was partitioned by Algorithm 1 into five subsets. The centers of the grey rectangles are marked with a cross. In each subset the member closest to the center is circled (a random member is circled in the subset with only two members). The representative set  $A_{(r)} = \{a_1, a_2, a_3, a_4, a_5\}$  is the set of all circled points.

Figure 3 illustrates the performance of PSA in selecting a subset from a randomly chosen (non-dominated) population in a three-objective space. A set of 500 randomly distributed points is depicted in Figure 3(a). The set is divided into 40 subsets, and the central member of each subset is selected as a representative point to form the representative subset depicted in Figure 3(f). According to Eq. (3), the diameter of the given set is the distance over  $f_2$ . Therefore, the first partition is made over  $f_2$ . At the second partition, the subset of the circles from Figure 3(b) has the largest dissimilarity and therefore is partitioned (over  $f_1$ ). At the next partition the subset of gray stars is partitioned over  $f_1$  to form the four subsets shown in Figure 3(d). The final stage of Algorithm 1 is shown in Figure 3(e). The subset shown in Figure 3(f) is obtained by selecting the point closest to the center of each



**Fig. 2** Selection of a representative subset  $A_{(r)}$  out of  $A$  using center point selection

of the 40 subsets. Figure 3(a) clearly shows that the distribution of the points in the original set is not uniform. Nevertheless, PSA managed to select a subset of fairly evenly distributed points from it.

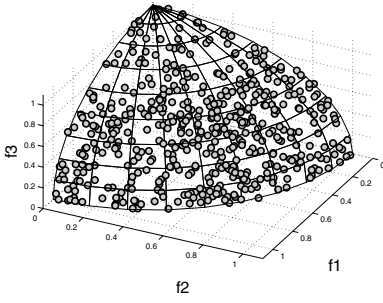
### 2.3 Complexity Analysis of the PSA

The following analysis refers to the procedure given in Algorithm 1. Let  $A = (\mathbf{f}_1, \dots, \mathbf{f}_n) \subset \mathbb{R}^k$  be the original set of  $n$  members, from which a subset of  $m$  members is to be selected.

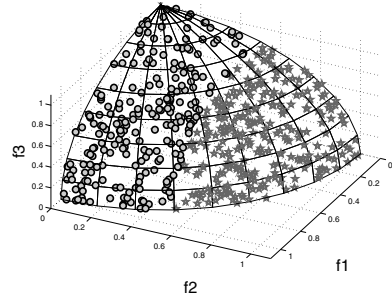
The evaluation of  $\varnothing A_1$  (Step 2) requires a comparison  $O(1)$  of  $n$  values to calculate  $\Delta_j$  for each of the  $k$  coordinates. The complexity of this step is  $O(nk)$ . To find  $\varnothing A_1 = \max_{j=1, \dots, k} \Delta_j$ ,  $k$  comparisons of  $O(1)$  are required. Thus, the total complexity of Step 2 is  $O(nk)$ . At Step 5  $i$  comparisons are required in order to find  $p_j$  and  $\varnothing A_j = \max_{l=1, \dots, i} \varnothing A_l$ , since it uses archiving. Given that  $i \leq m$ , the complexity of Step 5 is  $O(m)$  in the worst case. To partition the set  $A_j$  into two subsets  $A_{j1}$  and  $A_{j2}$ , each of the members of  $A_j$  has to be compared in coordinate  $p_j$  with  $a_j + \varnothing A_j/2$ . Since  $|A_j| = w_j \leq n$ , the complexity of Step 6 is  $O(n)$  in the worst case. The evaluation of  $\varnothing A_{j1}$  and  $\varnothing A_{j2}$  (Step 7) is similar to Step 2, but only  $w_j$  members have to be evaluated. In the worst case, the complexity of Step 7 is  $O(nk)$ . The complexity of Steps 5 to 7 is  $O(m) + O(n) + O(nk) = O(nk)$ . These steps are performed  $m - 1$  times. Therefore the entire complexity of Algorithm 1 is  $O(nkm)$ . The selection of one representative member of each set requires calculating the Euclidean distance of each member from the center of its set. The complexity of this stage is  $O(nk)$ . The comparison between the distances has a complexity  $O(n)$ .

We summarize the above considerations.

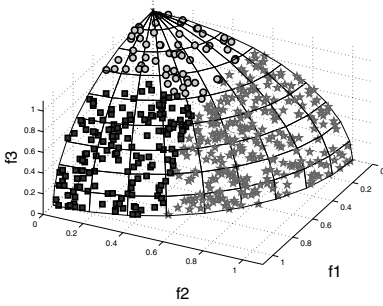
**Proposition 1** Given a set  $A = (\mathbf{f}_1, \dots, \mathbf{f}_n) \subset \mathbb{R}^k$ , the overall complexity of choosing  $m \leq n$  elements out of  $A$  using PSA and the center point selection is  $O(nkm)$ .



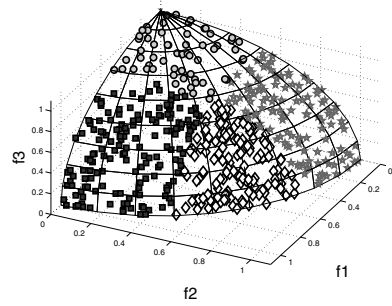
(a) original set of randomly distributed points



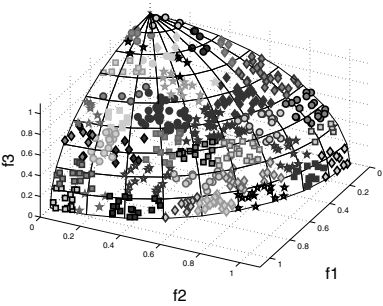
(b) two groups after first partition



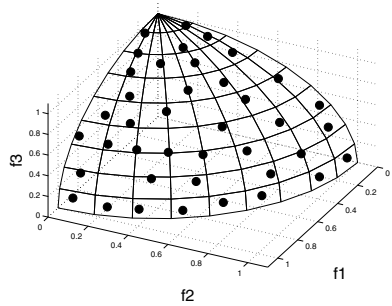
(c) three groups



(d) four groups



(e) final partition - 40 groups

(f) the selected subset  $A(r)$  containing the centers of the 40 groups

**Fig. 3** Demonstration of PSA in a three-dimensional space: Selection of a representative subset of 40 points from a randomly distributed set of 500 points

### 3 Integration of PSA into NSGA-II

Due to the minimal requirements of PSA—the algorithm needs only a set of candidate solutions and returns a subset of cardinality  $m$ —the algorithm can basically be integrated into any MOEA and used as a selection mechanism and/or as a crowding assignment mechanism. The advantages of such an integration are demonstrated in the following using the well-known NSGA-II [7] as a base MOEA. NSGA-II is commonly used for solving MOPS. Nevertheless, it is known for its incompetence of finding a well-diversified set of solutions for problems with more than two objectives. It is shown in Section 4 that the suggested integration improves this drawback, and is able to find an approximated set with a better diversity. The purpose of the following is not to introduce an algorithm which is better than the state-of-the-art, but to demonstrate the simplicity of improving an existing MOEA by integrating it with PSA.

The modified NSGA-II will be referred to as "NSGA2-PSA". For the sake of simplicity, the implementation will be described using the same notations as in [7], page 6, "The Main Loop". Hence, assume that the combined current population  $R_t = P_t \cup Q_t$  of the size  $2N$  is sorted to the non-dominated fronts  $\mathcal{F}_1, \dots, \mathcal{F}_l, \dots$  and

$$|\mathcal{F}_1| + \dots + |\mathcal{F}_{l-1}| = s < N, \quad |\mathcal{F}_1| + \dots + |\mathcal{F}_l| > N.$$

The next parent population  $P_{t+1}$  is constructed from all the members of the sets  $\mathcal{F}_1, \dots, \mathcal{F}_{l-1}$  and from  $N - s$  members of the set  $\mathcal{F}_l$ . The only difference from the classical NSGA-II is that instead of selecting  $N - s$  members from the last front  $\mathcal{F}_l$  according to the crowded-comparison operator  $\prec_n$ , selection is according to the PSA, as follows.  $\mathcal{F}_l$  is partitioned to  $N - s$  subsets according to Algorithm 1 and the central member of each subset is selected to  $P_{t+1}$ .

The crowding assignment of NSGA-II, used for the binary tournament is modified as well. Each set  $\mathcal{F}_i, i = 1, \dots, l$  is partitioned according to Algorithm 1 into  $\lfloor |\mathcal{F}_i|/2 \rfloor$  subsets, and every member of  $\mathcal{F}_i$  is assigned a crowding measure equal to the number of members of its subset.

To highlight the benefits of this integration, no other differences were made between NSGA-II and NSGA2-PSA. In particular, NSGA2-PSA uses the same tournament-selection, cross-over, and mutation operators as NSGA-II.

### 4 Comparison of NSGA2-PSA with NSGA-II

In this section, the proposed NSGA2-PSA is compared with the classic NSGA-II. For a fair comparison, both MOEAs start with the same initial population at every test, and use the same genetic settings.



## 4.1 Test Problems

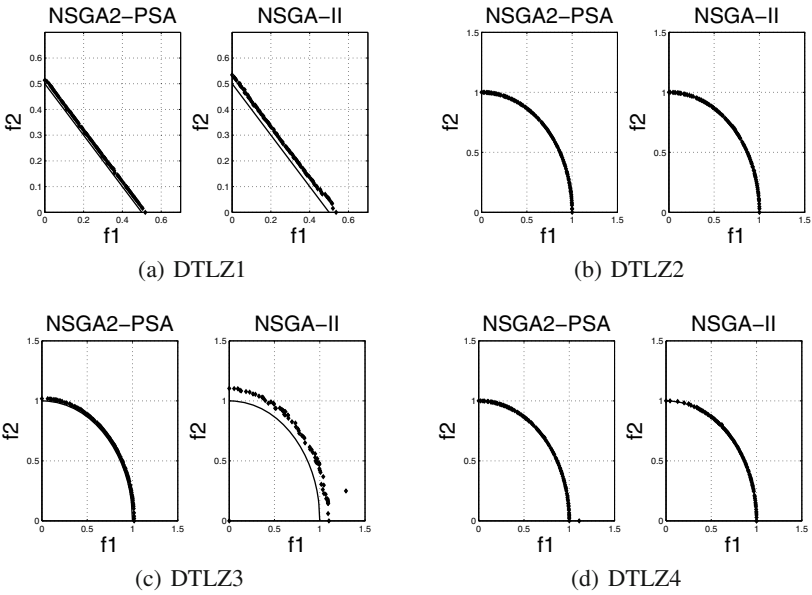
To demonstrate the scalability of the PSA, both algorithms are tested on the DTLZ 1 – 4 benchmark functions (see [8]). Each test case was tested for  $p \in \{2, 3, 4, 5\}$  objectives. The design space consists of  $d = 7$  decision variables for all problems. The Pareto optimal front (PF) of DTLZ1 is the hyper-plane of all points that satisfy the equation  $\sum f_i = 0.5, i = 1, \dots, M$ , where  $M$  is the number of objectives. This benchmark offers a difficult function for convergence with many local minima. The PF of DTLZ2 – 4 is the positive eighth of the hyper-sphere of all points that satisfy the equation  $\sum f_i^2 = 1, i = 1, \dots, M$ . DTLZ3 uses the same function for convergence as in DTLZ1, and therefore it can check the ability of a MOEA to find solutions close to the real PF. In DTLZ4 a very small section in the decision space is mapped to a very large section in the objective space. This property makes it a good benchmark to test the ability of a MOEA to find a diverse set of solutions. Since the proposed NSGA2-PSA aims to enhance the diversity of the approximated set, the results for the DTLZ4 are of major importance.

## 4.2 Genetic Settings

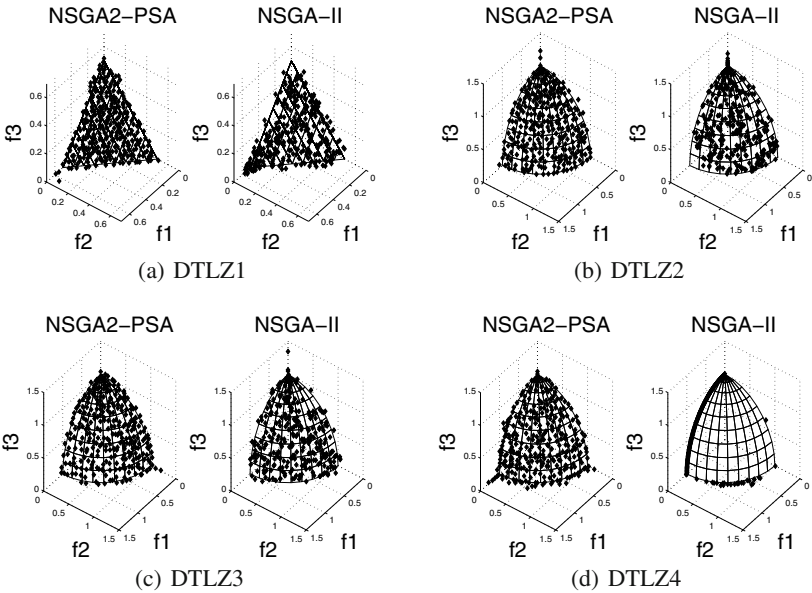
Both algorithms are given real-valued decision variables. They use the simulated binary crossover (SBX) operator and polynomial mutation [6], with distribution indexes of  $\eta_c = 15$  and  $\eta_m = 20$  respectively. A cross-over probability of  $p_c = 1$  and a mutation probability of  $p_m = 1/d$  are used. The population size is increased with the number of objectives. A population size of 100, 300, 600 and 1000 members is taken for 2, 3, 4 and 5 objectives, respectively. The number of generations for all the cases is set to 250.

## 4.3 Performance Metrics

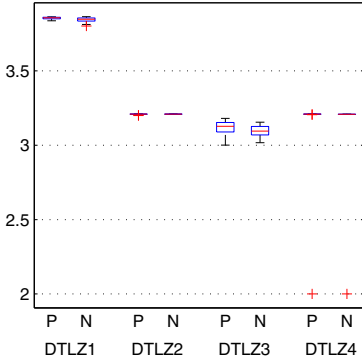
The approximated sets obtained by both methods are evaluated according to the hyper-volume [20], generational distance [23] and inverted generational distance [16]. The generational distance (GD) measures the average distance of the members in the final approximated set to the real PF. Therefore, GD measures the convergence of the algorithm. Since the mathematical definition of the fronts of the test-cases are known, the distances are calculated analytically. The inverted generational distance (IGD) measures the average distance from a set of well-diversified points on the real PF to the final approximated set. Since the reference set is well spread along the PF, IGD provides information about the diversity of the approximated set. Smaller values of GD and IGD are preferred. The hyper-volume (HV) measures the size of the region dominated by the approximated set. A higher value of HV is associated with a good spread and convergence.



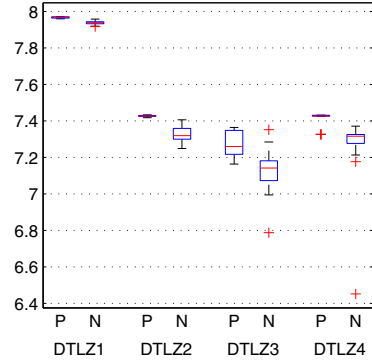
**Fig. 4** The final approximated set obtained by NSGA2-PSA and NSGA-II on two objectives DTLZ1 – DTLZ4



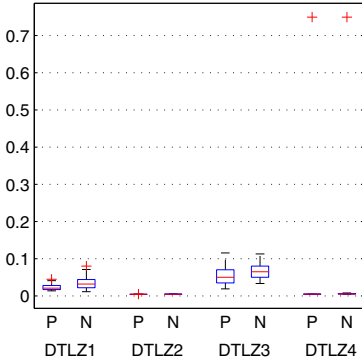
**Fig. 5** The final approximated set obtained by NSGA2-PSA and NSGA-II on three objectives DTLZ1 – DTLZ4



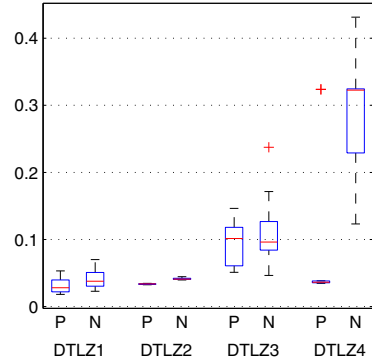
(a) HV results for 2 objectives



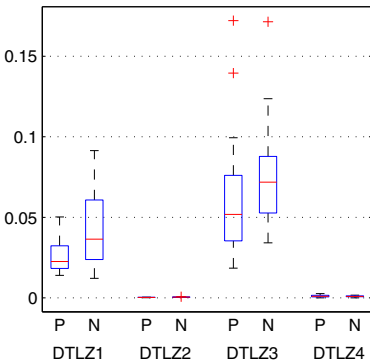
(b) HV results for 3 objectives



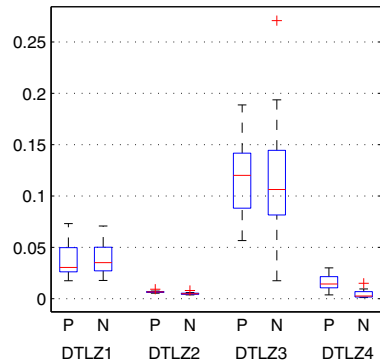
(c) IGD results for 2 objectives



(d) IGD results for 3 objectives

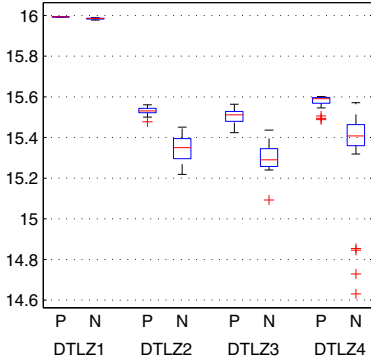


(e) GD results for 2 objectives

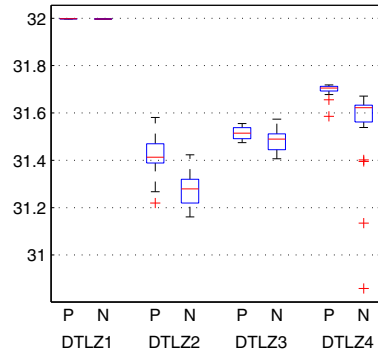


(f) GD results for 3 objectives

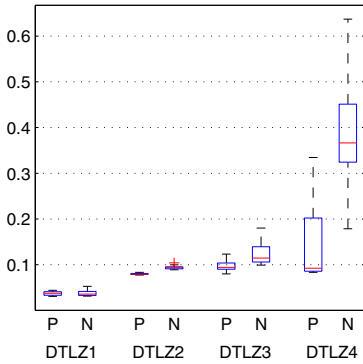
**Fig. 6** Box plots of results from 20 independent runs of NSGA-II (denoted as N) and NSGA2-PSA (denoted as P) on the DTLZ 1 – DTLZ 4 test cases for two and three objectives



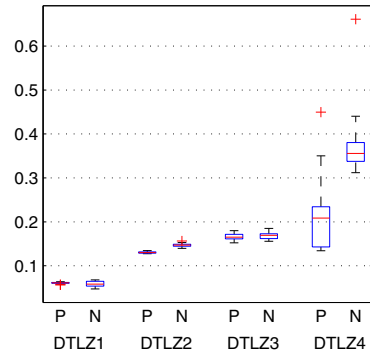
(a) HV results for 4 objectives



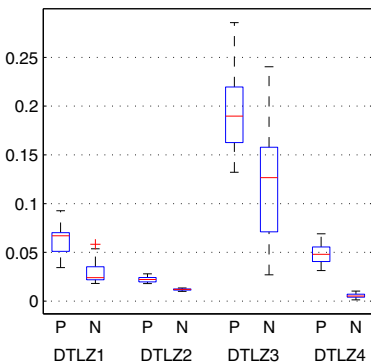
(b) HV results for 5 objectives



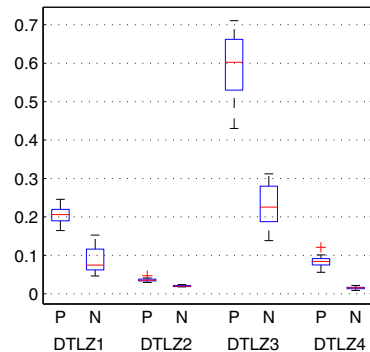
(c) IGD results for 4 objectives



(d) IGD results for 5 objectives



(e) GD results for 4 objectives



(f) GD results for 5 objectives

**Fig. 7** Box plots of results from 20 independent runs of NSGA-II (denoted as N) and NSGA2-PSA (denoted as P) on the DTLZ 1 – DTLZ 4 test cases for four and five objectives

#### 4.4 Simulation Results

To compare the methods, 20 independent runs were carried out on each of the problems. The results of a typical approximated set can be seen for two and three objectives in Figures 4 and 5 respectively. The statistical results for two – five objectives are shown in Figures 6 and 7. As seen in Figures 4 and 6(e), NSGA2-PSA has a slight advantage in convergence for problems with two objectives. That advantage can be seen in the results for the DTLZ1 and DTLZ3 problems. The differences between the methods begin to emerge for problems with three objectives and more. Figure 5 clearly shows that the spread of NSGA2-PSA is much better than that of NSGA-II for all test cases. The most outstanding difference occurs with the results of DTLZ4. As stated in [8], the final approximated set for DTLZ4 is highly dependent on the initial population. For three objectives, NSGA-II converged in some runs to the solutions on the f1-f2 plane, in others to the f1-f3 plane and in some runs it found solutions on the surface of the hyper-sphere. With the same initial population, NSGA2-PSA was able to find solutions on the surface of the hyper-sphere for almost every run. A good example of this can be seen in Figure 5(d).

The advantage of NSGA2-PSA over NSGA-II in spreading the approximated set is depicted in the statistical data in Figures 6 and 7. The IGD value of NSGA2-PSA is always better than that of NSGA-II. However, in problems with more than two objectives, this advantage comes at the cost of a loss of proximity, as manifested in the larger GD values of NSGA2-PSA. This highlights the unresolved tradeoff between proximity and diversity. The combination of small IGD values with large GD values can be explained by the existence of outliers. The higher HV measure of NSGA2-PSA indicates that the gain in diversity is more significant than the loss of proximity. The advantage of NSGA2-PSA makes the largest difference in the DTLZ4 test case, where a well-diversified set is hard to attain.

### 5 Conclusions and Future Work

In this paper, PSA (Part and Select Algorithm) has been proposed as a tool to select  $m$  elements from a given set  $A$  of candidate solutions such that the resulting subset  $A_{(r)}$  is characterized by high diversity (and hence serves as a good representation of  $A$ ). The idea of PSA was first used in [2] for function selection, but in this paper it is used and analyzed for the first time as a selection mechanism within MOEAs. The PSA together with center point selection is capable of choosing a well-spread subset  $A_{(r)}$  of  $A$  for any given value of  $m$ . To demonstrate the potential benefit of the novel algorithm, PSA has been integrated into the well-known NSGA-II. The purpose of this was to utilize the high competency of the partitioning algorithm to select a diversified sub-set from an arbitrary set for enhancing diversity along the Pareto front. The partitioning algorithm has been used to assign a crowding measure to each of the solutions, and was used within the evolutionary algorithm. It has been shown that the selection based on the partitioning algorithm evolves a

front involving higher diversity (based on IGD and on hypervolume). Nevertheless, the inherent trade-off between proximity and diversity was not resolved.

As future work, the potential of the partitioning algorithm should be tapped to select a good spread of solutions, e.g., by integrating it in other MOEAs such as BTS [3] or  $\Delta_p$ -EMOA [9]. The BTS algorithm aims directly at proximity and diversity while the  $\Delta_p$ -EMOA is aimed at Hausdorff approximations of the Pareto front (i.e., a performance measure that is also related to proximity and diversity [14]). Moreover, due to the advantages of PSA in choosing a representative subset, its potential for supporting decision-makers should be explored.

**Acknowledgements.** This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme. The last author acknowledges support from CONACyT project no. 128554.

The authors would like to thank Alan R. R. Freitas for his help with the complexity analysis.

## References

1. Ahn, C.W., Ramakrishna, R.S.: A diversity preserving selection in multiobjective evolutionary algorithms. *Applied Intelligence* 32(3), 231–248 (2010), doi:10.1007/s10489-008-0140-0
2. Avigad, G., Goldvard, A., Salomon, S.: Time-response-based evolutionary optimization. Tech. Rep. br352012, Ort Braude Academic College (2012), <http://brd.braude.ac.il/gideon>
3. Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 7(2), 174–188 (2003)
4. Coello Coello, C.A., van Veldhuizen, D.A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer (2007)
5. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York (2001)
6. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Syst.* 9, 115–148 (1995)
7. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimisation: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000), <http://dl.acm.org/citation.cfm?id=645825.668937>
8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. Report 2 001 001, Kanpur Genetic Algorithms Lab (KANGAL), Indian Inst. Technol., Kanpur, India (2001)
9. Gerstl, K., Rudolph, G., Schütze, O., Trautmann, H.: Finding evenly spaced fronts for multiobjective control via averaging Hausdorff-measure. In: *Int'l. Proc. Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2011)*, pp. 975–980 (2011)

10. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched Pareto genetic algorithm for multi-objective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pp. 82–87. IEEE Press (1994)
11. Laumanns, M., Očenášek, J.: Bayesian Optimization Algorithms for Multi-objective Optimization. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 298–307. Springer, Heidelberg (2002)
12. Li, M., Zheng, J., Xiao, G.: An efficient multi-objective evolutionary algorithm based on minimum spanning tree. In: Evolutionary Computation, CEC 2008 (IEEE World Congress on Computational Intelligence), pp. 617–624 (2008), doi:10.1109/CEC.2008.4630860
13. Wineberg, M., Oppacher, F.: The Underlying Similarity of Diversity Measures Used in Evolutionary Computation. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1493–1504. Springer, Heidelberg (2003)
14. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation 1 (to appear, 2012), doi:10.1109/TEVC.2011.2161872
15. Shen, X., Zhang, M., Li, T.: A multi-objective optimization evolutionary algorithm addressing diversity maintenance. In: International Joint Conference on Computational Sciences and Optimization, CSO 2009, vol. 1, pp. 524–527 (2009), doi:10.1109/CSO.2009.28
16. Sierra, M.R., Coello, C.A.C.: Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and  $\epsilon$ -Dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 505–519. Springer, Heidelberg (2005)
17. Solow, A., Polasky, S.: Measuring biological diversity. *Environmental and Ecological Statistics* 1, 95–103 (1994), <http://dx.doi.org/10.1007/BF02426650>, doi:10.1007/BF02426650
18. van Veldhuizen, D.A.: Multiobjective evolutionary algorithms: Classifications, analyses and new innovations. Ph.D. thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA (1999)
19. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007), doi:10.1109/TEVC.2007.892759
20. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph.D. thesis, ETH, Zurich, Switzerland (1999)
21. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
22. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K.C., Tsahalis, D.T., Périaux, J., Papailiou, K.D., Fogarty, T. (eds.) Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems. International Center for Numerical Methods in Engineering, Athens, Greece, pp. 95–100 (2001)

# The Gradient Free Directed Search Method as Local Search within Multi-Objective Evolutionary Algorithms

Adriana Lara, Sergio Alvarado, Shaul Salomon, Gideon Avigad,  
Carlos A. Coello Coello, and Oliver Schütze

**Abstract.** Recently, the Directed Search Method has been proposed as a point-wise iterative search procedure that allows to steer the search, in any direction given in objective space, of a multi-objective optimization problem. While the original version requires the objectives' gradients, we consider here a possible modification that allows to realize the method without gradient information. This makes the novel algorithm in particular interesting for hybridization with set oriented search procedures, such as multi-objective evolutionary algorithms.

In this paper, we propose the DDS, a gradient free Directed Search method, and make a first attempt to demonstrate its benefit, as a local search procedure within a memetic strategy, by integrating the DDS into the well-known algorithm MOEA/D. Numerical results on some benchmark models indicate the advantage of the resulting hybrid.

## 1 Introduction

Many real world problems demand for the concurrent optimization of  $k$  objectives leading to a *multi-objective optimization problem* (MOP) [16]. One characteristic of these problems, compared with those where only *one* objective is under consideration, is that the solution set of a MOP (the *Pareto set*) typically forms a

---

Adriana Lara

Mathematics Department ESFM-IPN, Edif. 9 UPALM, 07300 Mexico City, Mexico  
e-mail: [adriana@esfm.ipn.mx](mailto:adriana@esfm.ipn.mx)

Sergio Alvarado · Carlos A. Coello Coello · Oliver Schütze  
Computer Science Department, CINVESTAV-IPN, Av. IPN 2508,  
Col. San Pedro Zacatenco, 07360 Mexico City, Mexico  
e-mail: [ccoello,schuetze@cs.cinvestav.mx](mailto:ccoello,schuetze@cs.cinvestav.mx),  
[salvarado@computacion.cs.cinvestav.mx](mailto:salvarado@computacion.cs.cinvestav.mx)

Shaul Salomon · Gideon Avigad  
ORT Braude School of Engineering, Karmiel, Israel  
e-mail: [shaulsal,gideon@braude.ac.il](mailto:shaulsal,gideon@braude.ac.il)



$(k - 1)$ -dimensional object. So far, there exist many methods for the computation of the Pareto set of a MOP. Among them, multi-objective evolutionary algorithms (MOEAs) have caught the attraction of many researchers (e.g., [7, 6] and references therein). The major reason for this might be that the population based approach, together with a stochastic component in the search procedure, allows typically for an approximation of the entire (global) Pareto set in one single run of the algorithm. This represents an advantage over most mathematical programming (MP) techniques, which require in addition certain smoothness assumptions on the MOP. On the other hand, it is well-known that MOEAs normally need a large amount of function evaluations, due to their slow convergence rate, in order to generate a suitable finite size approximation of the set of interest ([4]). As a remedy, researchers have proposed *memetic MOEAs*, i.e., hybrids of MOEAs and MP with the aim to get fast and reliable global search procedures (e.g., [9, 11, 10, 22, 14]).

In this paper, we adapt the Directed Search (DS) method [21] for the use within MOEAs. One crucial drawback of the DS is that it requires gradient information which restricts its usability. Here, we propose a modification of the DS that is gradient free. Even more, the computation of the search direction comes without the cost of additional function evaluations if the neighborhood information can be exploited. The latter makes the Discrete Directed Search (DDS) a suitable algorithm, in particular, for the usage within set oriented search techniques. We demonstrate the benefit of the DDS by hybridizing it with MOEA/D ([24]), a state-of-the-art MOEA whose neighborhood definition can be directly used for the DDS.

The remainder of this paper is organized as follows: In Section 2, we give the required background for the understanding of the sequel. In Section 3, we present the DDS, a gradient free Directed Search variant. In Section 4, we propose a way to integrate the DDS into MOEA/D leading to a new memetic algorithm. In Section 5, we present some results, and finally, we draw our conclusions in Section 6.

## 2 Background

In the following we consider unconstrained multi-objective optimization problems (MOPs) which can be stated as follows:

$$\min_{x \in \mathbb{R}^n} F(x), \quad (1)$$

where  $F : R \subset \mathbb{R}^n \rightarrow \mathbb{R}^k$  is defined as the vector of  $k$  objective functions  $f_i : R \subset \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, k$ . A point  $x \in R$  is said to dominate another point  $y \in R$ , if  $f_i(x) \leq f_i(y)$  for all  $i \in \{1, \dots, k\}$ , and if there exists an index  $j \in \{1, \dots, k\}$  such that  $f_j(x) < f_j(y)$ . A point  $x \in R$  is called optimal, or *Pareto optimal*, with respect to (1), if there is no other point  $y \in R$  that dominates  $x$ . The set of all optimal solutions is called the Pareto set, and the set of images of the optimal solutions is called the Pareto front.

Recently, a numerical method has been proposed for differentiable MOPs that allows to steer the search from a given point into a desired direction  $d \in \mathbb{R}^k$  in

objective space ([21]). To be more precise, given a point  $x \in \mathbb{R}^n$ , a search direction  $v \in \mathbb{R}^n$  is sought such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + tv) - f_i(x_0)}{t} = d_i, \quad i = 1, \dots, k. \quad (2)$$

Such a direction vector  $v$  solves the following system of linear equations:

$$J(x_0)v = d, \quad (3)$$

where  $J(x)$  denotes the Jacobian of  $F$  at  $x$ . Since typically  $k \ll n$ , we can assume that the system in Equation (3) is (highly) underdetermined. Among the solutions of Equation (3), the one with the smaller 2-norm can be viewed as the greedy direction for the given context. This solution is given by

$$v_+ := J(x)^+ d, \quad (4)$$

where  $J(x)^+$  denotes the pseudo inverse of  $J(x)$  (we refer e.g. to [17] for an efficient computation of  $v_+$ ). If one proceeds the search in direction  $d$  in the same manner, this is identical to the numerical solution of the following initial value problem (starting from solution  $x_0 \in \mathbb{R}^n$ ):

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= v_+(x(t)), \quad t > 0 \end{aligned} \quad (5)$$

If  $d$  is a ‘descent direction’ (i.e.,  $d_i \leq 0$  for all  $i = 1, \dots, k$  and there exists an index  $j$  such that  $d_j < 0$ ), a numerical solution of (5) can be viewed as a particular hill climber for MOPs.

The endpoint  $x^*$  of the solution curve of (5) does not necessarily have to be a Pareto point, but it is a boundary point in objective space, i.e.,  $F(x^*) \in \partial F(\mathbb{R}^n)$  which means that the gradients of the objectives in  $x^*$  are linear independent (and hence, that  $\text{rank}(J(x^*)) < k$ ). This fact can be used to check numerically if a current iterate is near to a boundary point: For the condition number of the Jacobian it holds

$$\kappa_2(J(x)) = \sqrt{\frac{\lambda_{\max}(J(x)^T J(x))}{\lambda_{\min}(J(x)^T J(x))}} \rightarrow \infty \quad \text{for } x \rightarrow x^*, \quad (6)$$

where  $\lambda_{\max}(A)$  and  $\lambda_{\min}(A)$  denote the largest and the smallest eigenvalue of matrix  $A$ , respectively. (Roughly speaking, the condition number indicates how ‘near’ the rows of  $J(x)$ , i.e., the gradients of the objectives, are to be linearly independent: the higher the value of  $\kappa_2(J(x))$ , the closer  $J(x)$  is to a matrix with rank less than  $k$ .) Further, one can check the (approximated) endpoint  $x^*$  numerically for optimality by checking if  $\|\sum_{i=1}^k \tilde{\alpha}_i \nabla f_i(x^*)\|_2 \leq \text{tol}$ , where  $\text{tol} > 0$  is a given tolerance and  $\tilde{\alpha}$  solves the following  $k$ -dimensional quadratic optimization problem (see [18]):

$$\min_{\alpha} \left\{ \left\| \sum_{i=1}^k \alpha_i \nabla f_i(x) \right\|_2^2 : \alpha_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \alpha_i = 1 \right\} \quad (7)$$

The hill climber described above shares many characteristics with the one described in [3], where also possible choices for  $d$  are discussed.

### 3 Gradient Free Directed Search

The key of the DS is to solve Equation (3) in order to find a vector  $v$  such that the search can be steered in  $d$ -direction. For this, the most expensive part might be the computation or approximation of the objectives' gradients. Here, we suggest an alternative way to compute such search directions  $v$  using a finite difference method tailored to the given context. We note that this approach is not equal to the classical finite difference approach used to approximate the gradient (e.g., [17]).

Assume we are given a candidate solution  $x \in \mathbb{R}^n$  and  $r$  search directions  $v_i \in \mathbb{R}^n$ ,  $i = 1, \dots, r$ . Define the matrix  $\mathcal{F}(x) \in \mathbb{R}^{k \times r}$  as follows:

$$\mathcal{F}(x) := (\langle \nabla f_i(x), v_j \rangle) \quad i = 1, \dots, k; \quad j = 1, \dots, r \quad (8)$$

That is, every entry  $m_{ij}$  of  $\mathcal{F}$  is defined by the directional derivative of objective  $f_i$  in direction  $v_j$ ,  $m_{ij} = \nabla_{v_j} f_i(x)$ . Crucial for the subsequent discussion is the following result:

**Proposition 1** *Let  $x, v_i, i = 1, \dots, r \in \mathbb{R}^n$ ,  $\lambda \in \mathbb{R}^r$ , and  $v := \sum_{i=1}^r \lambda_i v_i$ . Then*

$$J(x)v = \mathcal{F}(x)\lambda \quad (9)$$

*Proof.* It is

$$\mathcal{F}(x)\lambda = \begin{pmatrix} \langle \nabla f_1(x), v_1 \rangle & \dots & \langle \nabla f_1(x), v_r \rangle \\ \vdots & \vdots & \vdots \\ \langle \nabla f_k(x), v_1 \rangle & \dots & \langle \nabla f_k(x), v_r \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} \quad (10)$$

and

$$J(x)v = J(x) \left( \sum_{i=1}^r \lambda_i v_i \right) = \sum_{i=1}^r \lambda_i \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix} v_i \quad (11)$$

Hence, for the  $l$ -th component of both products it holds

$$(\mathcal{F}(x)\lambda)_l = \sum_{i=1}^r \lambda_i \langle \nabla f_i(x), v_i \rangle = (J(x)v)_l, \quad (12)$$

and the desired identity follows.  $\square$

Hence, in search for a direction  $v$ , one can instead of Equation (3) try to solve the following equation:

$$\mathcal{F}(x)\lambda = d, \quad (13)$$

and set

$$v := \sum_{i=1}^r \lambda_i v_i. \quad (14)$$

**Remark 1** Assume that we are given a candidate solution  $x_0 \in \mathbb{R}^n$  and further  $r$  points  $x_i$ ,  $i = 1, \dots, r$ , in the neighborhood of  $x_0$  together with their function values  $F(x_i)$ ,  $i = 0, \dots, r$ . Defining

$$v_j := \frac{x_j - x_0}{\|x_j - x_0\|_2}, \quad t_j := \|x_j - x_0\|_2, \quad j = 1, \dots, r, \quad (15)$$

one can approximate the entries of  $\mathcal{F}$  by finite differences as follows:

$$\begin{aligned} m_{ij} &= \langle \nabla f_i(x_0), v_j \rangle = \lim_{t \searrow 0} \frac{f_i(x_0 + tv_j) - f_i(x_0)}{t} \\ &\approx \frac{f_i(x_j) - f_i(x_0)}{\|x_j - x_0\|_2}, \quad i = 1, \dots, k, \quad j = 1, \dots, r. \end{aligned} \quad (16)$$

Analog to the well-known forward differences to approximate the gradient, one can show that the computational error is given by

$$\langle \nabla f_i(x_0), v_j \rangle = \frac{f_i(x_j) - f_i(x_0)}{\|x_j - x_0\|_2} + O(\|x_j - x_0\|_2). \quad (17)$$

Note that, by this, the search direction can be computed without any additional function evaluations.

Since it is ad hoc not clear if Equation (13) has a solution, and even if it is solvable, how the condition of the problem is (in terms of  $\kappa_2(\mathcal{F})$ ), we have to investigate the choice of  $r$  and the  $v_i$ 's. For this, it is advantageous to write  $\mathcal{F}(x)$  as follows:

$$\mathcal{F}(x) = J(x)V, \quad (18)$$

where  $V := (v_1, \dots, v_r) \in \mathbb{R}^{n \times r}$  is the matrix consisting of the search directions  $v_i$ . If  $\text{rank}(J(x)) = k$  (which is given for a non-boundary point  $x$ ), it is known from linear algebra that

$$\text{rank}(J(x)) = k \quad \Rightarrow \quad \text{rank}(\mathcal{F}(x)) = \text{rank}(V). \quad (19)$$

If on the other hand  $x$  is a boundary point (and hence,  $\text{rank}(J(x)) < k$ ), then it follows by the rank theorem of matrix multiplication that also  $\text{rank}(\mathcal{F}(x)) < k$  regardless of the choice of  $V$  (i.e., regardless of the number  $r$  and the choice of the search directions  $v_i$ ).

This indicates that the condition number of  $\mathcal{F}(x)$  can be used to check numerically if a current iterate is already near to an endpoint of (5). Equation (19) indicates that the  $v_i$ 's should be chosen such that they are linearly independent. If in addition the search directions are orthogonal to each other, a straightforward calculation shows that

$$V \text{ orthogonal} \Rightarrow \kappa_2(\mathcal{F}(x)) = \kappa_2(J(x)). \quad (20)$$

In that case, the condition number  $\kappa_2(\mathcal{F}(x))$  can indeed be used as a stopping criterion, analog to the original method described in Section 2. That is, one can stop the iteration if for a current iterate  $x_i$  it holds

$$\kappa_2(\mathcal{F}(x_i)) > \text{tol}_\kappa, \quad (21)$$

where  $\text{tol}_\kappa \gg 1$  is a large number.

**Example 1** Consider the following bi-objective model ([12]):

$$\begin{aligned} F : \mathbb{R}^n &\rightarrow \mathbb{R}^2 \\ f_i(x) &= \|x - a_i\|_2^2, \quad i = 1, 2, \end{aligned} \quad (22)$$

where  $a_1 = (1, \dots, 1)^T, a_2 = (-1, \dots, -1)^T \in \mathbb{R}^n$ . The Pareto set is given by the line segment between  $a_1$  and  $a_2$ , i.e.,

$$\mathcal{P} = \{x \in \mathbb{R}^n : x_i = 2\alpha - 1, i = 1, \dots, k, \alpha \in [0, 1]\} \quad (23)$$

Let  $r = 2$  and  $v_1 := e_i$  and  $v_2 := e_j, i \neq j$ , where  $e_i$  denotes the  $i$ -th canonical vector. Then, it is

$$\mathcal{F}(x) = \begin{pmatrix} x_i - 1 & x_j - 1 \\ x_i + 1 & x_j + 1 \end{pmatrix} \quad (24)$$

It is  $\det(\mathcal{F}(x)) = 1/(2(x_i - x_j))$ , and hence,

$$\det(\mathcal{F}(x)) = 0 \Leftrightarrow x_i = x_j, \quad (25)$$

by which it follows that it is  $\text{rank}(\mathcal{F}(x)) = 2$  for all  $x \in \mathbb{R}^n \setminus B$ , where  $B := \{x \in \mathbb{R}^n : x_i = x_j\}$  (note that  $\mathcal{P} \subset B$ ). Since  $B$  is a zero set in  $\mathbb{R}^n$ , the probability is one that for a randomly chosen point  $x \in \mathbb{R}^n$  the matrix  $\mathcal{F}(x)$  has full rank, and hence, that Equation (13) has a unique solution. To be more precise, it is  $v = \lambda_1 e_i + \lambda_2 e_j$ , where

$$\begin{aligned} \lambda &= \mathcal{F}^{-1}(x)d = \frac{1}{\det(\mathcal{F}(x))} \begin{pmatrix} x_{j+1} & -x_j + 1 \\ -x_i - 1 & x_j - 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \\ &= \frac{1}{2(x_i - x_j)} \begin{pmatrix} x_j(d_1 - d_2) + d_1 + d_2 \\ x_i(d_2 - d_1) - d_1 - d_2 \end{pmatrix}. \end{aligned} \quad (26)$$

Note that this holds regardless of the number  $n \geq 2$  of the parameter dimension.

The above considerations show that already for  $r = k$  search directions  $v_i$ ,  $i = 1, \dots, r$ , one can find a descent direction  $\tilde{v}$  by solving Equation (13). However, by construction it is  $v \in \text{span}\{v_1, \dots, v_k\}$  which means that only a  $k$ -dimensional subspace of the  $\mathbb{R}^n$  is explored in one step. One would expect that the more search directions  $v_i$  are taken into account, the better the choice of  $\tilde{v}$  is. This is indeed the case: For  $r > k$ , we suggest to choose analog to (4)

$$v_+^{(r)} := \sum_{i=1}^r \lambda_i v_i, \quad \text{where } \lambda = \mathcal{F}(x_0)^+ d \quad (27)$$

The following discussion gives a relation between  $v_+^{(r)}$  and  $v_+$  for non-boundary points  $x$  for the case that the  $v_i$ 's are orthonormal: It is

$$v_+ = J^+(x)d = J(x)^T (J(x)J(x)^T)^{-1}d \quad (28)$$

and

$$\begin{aligned} \lambda &= \mathcal{F}(x)^+ d = V^T J(x)^T (J(x) \underbrace{V V^T}_{I} J(x)^T)^{-1} d \\ &= V^T \underbrace{J(x)^T (J(x)J(x)^T)^{-1}}_{v_+} d = V^T v_+ \end{aligned} \quad (29)$$

and hence

$$v_+^{(r)} = \sum_{i=1}^r \lambda_i v_i = \sum_{i=1}^r \langle v_i, v_+ \rangle v_i \quad (30)$$

For instance, when choosing  $v_i = e_{j_i}$ , Equation (30) gets simplified:

$$v_+^{(r)} = \sum_{i=1}^r v_{+,j_i} e_{j_i}, \quad (31)$$

i.e.,  $v_+^{(r)}$  has only  $r$  entries which are identical to the corresponding entries of  $v_+$ . In both cases  $v_+^{(r)}$  gets closer to  $v_+$  with increasing number  $r$  and for  $r = n$  it is  $v_+^{(r)} = v_+$ .

**Remark 2** *We would like to stress that this approach is intended for multi-objective optimization problems (i.e.,  $k > 1$ ). For the special (and important) case of scalar optimization (i.e.,  $k = 1$ ), the present approach is of very limited value as the following discussion shows: For  $r = k = 1$ , Equation (13) reads as*

$$\langle \nabla f(x), v_1 \rangle \lambda = d \quad (32)$$

*Concrete values for the desired direction  $d$  in image space are hard to find. If it is e.g. desired to obtain improvements of the objective  $f$ , one may choose (after normalization)  $d = -1$ . Assuming that  $\langle \nabla f(x), v_1 \rangle \neq 0$ , Equation (32) leads then (again after normalization) to*

$$\lambda = \begin{cases} -1 & \text{if } \langle \nabla f(x), v_1 \rangle > 0 \\ 1 & \text{if } \langle \nabla f(x), v_1 \rangle < 0 \end{cases} \quad (33)$$

and thus to  $v \in \{v_1, -v_1\}$ . However, this does not bring any new insight: It is well-known that the descent cone of  $f$  at  $x$  is given by

$$C(x) := \{v \in \mathbb{R}^n : \langle \nabla f(x), v \rangle < 0\}, \quad (34)$$

and hence, it is under the above assumption on  $v_1$  either  $v_1 \in C(x)$  or  $-v_1 \in C(x)$ .

Finally, we state the Discrete Directed Search (DDS) which is simply a line search along the search direction  $v_+^{(r)}$  (see Algorithm 1). For an efficient step size control we refer to [15].

---

#### Algorithm 1 Discrete Directed Search (DDS)

---

**Require:** Initial solutions  $x_0, x_1, \dots, x_r \in \mathbb{R}^n$

**Ensure:** New candidate solution  $x_{new}$

1: compute  $v_+^{(r)}$  as in Eq. (27).

2: compute  $t \in \mathbb{R}_+$

3:  $x_{new} := x_0 + t v_+^{(r)}$

---

## 4 Integration of DDS into MOEA/D

Here we show the potential of the DDS as local search engine within the state-of-the-art method MOEA/D [24]. The philosophy behind this MOEA consists of employing a decomposition approach, to convert the problem of approximating the Pareto front into a certain number of scalar optimization problems (SOPs). We stress that MOEA/D is indeed particularly attractive to be combined with the DDS procedure. Two important reasons for this are (a) MOEA/D has an implicit neighborhood structure, imposed by the particular decomposed problems, and (b) there is a weight vector associated to each subproblem, and to each individual.

In this sense, DDS can take advantage of (a) to avoid the computation of the neighbors used to estimate the search direction  $v_+^{(r)}$ ; also, no extra function evaluations are necessary, which makes the computation of the search direction an effortless procedure—in terms of function evaluations. In other words, given a point  $x \in \mathbb{R}^n$ , if some neighbors of  $x_1, \dots, x_r$  are already evaluated, the computation can be done without any additional function evaluations. In general, memetic MOEAs which use gradient-based information have already proven their efficacy on several MOPs [4, 3], but the cost of estimating the first order information has been always an issue.

From a practical point of view, the reason (b) allows us to automatically identify which is the individual with the best fitness associated with each subproblem; and establishing, in this manner, a relationship with the corresponding weight vector

for the movement performed by DDS. Furthermore, DDS also takes advantage that MOEA/D already has a computed reference point for the decomposed problems. In this sense, once the individual  $p$  is chosen to be affected by the local search, the values for  $d_i$  in the DDS are already set for  $p$  (according with its corresponding decomposed MOEA/D subproblem).

How to apply the local search is one of the main issues when designing memetic algorithms. Two important parameters have been identified [8, 13] as crucial when controlling the local search application on memetic MOEAs. They are:

- (i) The frequency  $k_{ls}$  for application of the local search along the total amount of generations.
- (ii) The number of elements  $h_{ls}$ , from the population, to which the local search is applied each generation.

Algorithm 2 describes the coupling of DDS and MOEA/D. The notation regarding MOEA/D procedures and parameters is consistent with the one presented in [24]. The SOP regarding the decomposition was, in this case, taken by the Tchebycheff approach as:

$$\text{minimize } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i | f_i(x) - z_i^*\} \quad (35)$$

where  $z^*$ , such that  $z_i^* = \min\{f_i(x) | x \in P_0\}$ , is the reference point; and the direction  $d_i$  for the application of DDS to the individual  $x_i$  is set as  $d_i = \lambda_i - z^*$ .

---

### Algorithm 2 MOEA/D/DDS

---

- 1: Set the weight vectors  $\lambda_i$  and the neighborhoods  $B(i) = \{i_1, \dots, i_T\}$  for each decomposed problem ( $\lambda_{i_1}, \dots, \lambda_{i_T}$  are the  $T$  closest weight vectors to  $\lambda_i$ ).
  - 2: Initialize an initial population  $P_0 = \{x_1, \dots, x_N\}$ .
  - 3: Initialize the reference point  $z^*$ ,  $EP = \emptyset$ ,  $gen = 1$ .
  - 4: **repeat**
  - 5:   **for**  $i = 1, \dots, N$  **do**
  - 6:     Select two indexes  $k, l$  from  $B(i)$  and generate, using genetic operators, a new solution  $y$  from  $x_k$  and  $x_l$ .
  - 7:     Apply the subproblem improvement heuristic for each  $y$  in order to get  $y'$  (Eq. 35).
  - 8:     **if**  $mod(gen, k_{ls}) == 0$  and  $mod(i, h_{ls}) == 0$  **then**
  - 9:       Apply DDS to  $y'$ , in order to get  $y''$ .
  - 10:       Set  $y' \leftarrow y''$ .
  - 11:     **end if**
  - 12:     Update the reference point  $z^*$ .
  - 13:     Remove from  $EP$  all the vectors dominated by  $y'$  and add it if no vectors in  $EP$  dominate  $y'$ .
  - 14:   **end for**
  - 15:    $gen = gen + 1$ .
  - 16: **until** Stopping criteria is satisfied
  - 17: report  $EP$ .
-



## 5 Numerical Results

In this section we show some results of the MOEA/D/DDS for the computation of Pareto fronts as well as of the DS in the context of a particular control problem.

### 5.1 Comparison MOEA/D and MOEA/D/DDS

Since we have chosen MOEA/D as base MOEA, it seems reasonable to test over the CEC09 benchmark [27]. For this, we adapted the available code from a specific version of MOEA/D [25], which was tested for performance with remarkable results over this particular test suite [26]. Differences of this code and the MOEA/D original version are that this modification allows the computational effort to be distributed among the subproblems based on an utility function  $\pi_i$  defined for each subproblem.

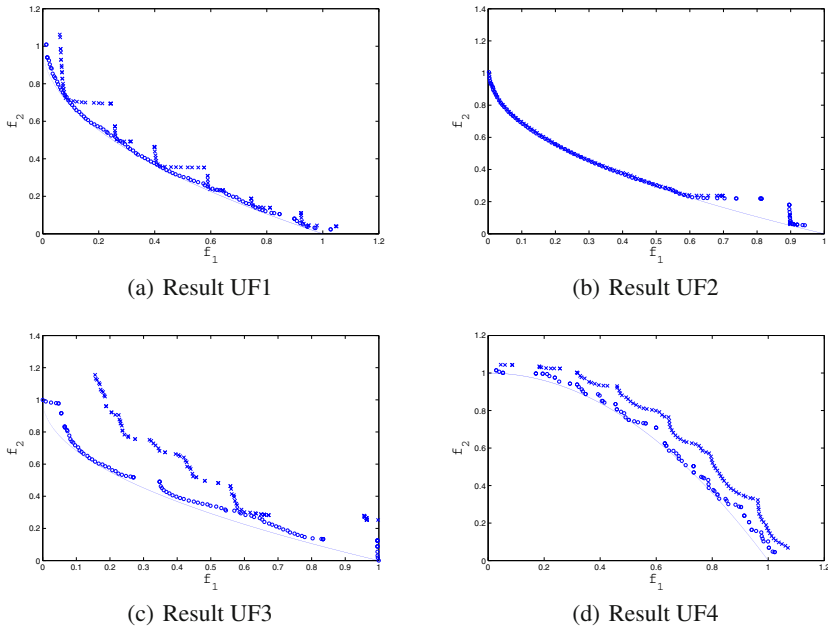
The main parameters for MOEA/D were set according to Table 1, and for the DDS we have chosen  $r = 5$ . We stop the computations after 30,000 function evaluations, which represents the 10% of the budget originally allowed by the competition. Figure 1 presents plots that show that the Pareto front has been reached, by the MOEA/D/DDS using this reduced budget. Finally, the parameters related to the control for application of the local search are presented in Table 2. As performance indicators to compare the results of the different algorithms we have chosen to take the Generational Distance (GD, see [23]), the Inverted Generational Distance (IGD, see [5]), the averaged Hausdorff Distance  $\Delta_1$  (see [19, 20]) which is in fact the maximum of the GD and the IGD value, and the Hypervolume indicator (HV, see [28]). From Figure 1 and Table 3 it becomes clear that the new hybrid is outperforming its base MOEA in three out of four cases. For UF2, the indicator values of MOEA/D are slightly better, however, there is no clear winner.

**Table 1** Parameters setting for MOEA/D in this experiments

Identifier	Value	Description
N	600	The number of subproblems considered
T	0.1 N	Size of the neighborhood
$P_m$	1/n	Mutation rate
EP	100	Number of final solutions (external population)

**Table 2** Parameters setting for the memetic part

Identifier	Value	Description
$k_{ls}$	(0.15) tg	Local search application frequency; tg is the total number of generations.
$h_{ls}$	(0.1) N	Percentage of the population over which the local search is applied.



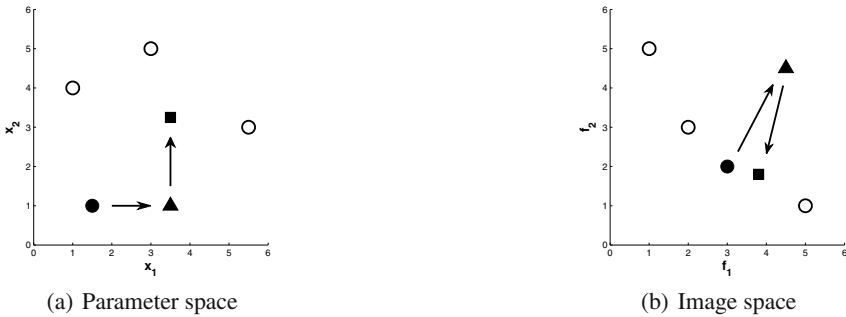
**Fig. 1** Numerical results for MOEA/D (crosses) and MOEA/D/DDS (circles) on the benchmark models UF1 to UF4 (compare also to Table 3). The true Pareto fronts are indicated by the dotted lines.

**Table 3** Indicator values obtained by MOEA/D and MOEA/D/DDS on the benchmark models UF1 to UF4. The budget for the function evaluations was set to 30,000. The information was gathered by 10 independent runs.

Problems		Indicators			
		<i>GD</i>	<i>IGD</i>	$\Delta_1$	HV
UF1	MOEA/D	0.0696046570	0.0709107497	0.0736649773	0.9431136545
	MOEA/D/DDS	<b>0.0415884479</b>	<b>0.0400041235</b>	<b>0.0422788829</b>	<b>0.9619027643</b>
UF2	MOEA/D	<b>0.0261457333</b>	<b>0.0195839746</b>	<b>0.0261457333</b>	<b>0.9757865917</b>
	MOEA/D/DDS	0.0323564751	0.0151025484	0.0323564751	0.9645998451
UF3	MOEA/D	0.1459679411	0.1307335754	0.1520909424	0.8604675543
	MOEA/D/DDS	<b>0.0552610854</b>	<b>0.0537723289</b>	<b>0.0616385221</b>	<b>0.9608503276</b>
UF4	MOEA/D	0.0823081769	0.0871159952	0.0871159952	0.9206159284
	MOEA/D/DDS	<b>0.0472797997</b>	<b>0.0472797997</b>	<b>0.0478409975</b>	<b>0.9498433991</b>

## 5.2 A Control Problem

The skill of the DS is to steer the search into any direction in objective space. This can be used for Pareto front computations as seen above, however, may also have other applications as the following discussion shows. In [11], robustness of optimal solutions to MOPs subject to physical deterioration has been addressed. The problem posed in that study involves the need to steer the decorated performances (due to undesired changes in some design parameters) as close as possible to the original performances. In order to elucidate this demand for robustness, consider a two parameter bi-objective design space (i.e.,  $n = k = 2$ ). Assume Figure 2 shows four optimal solutions, and that the performance vector designated by the bold circle is the decision makers selected solution (denote by  $x^*$ ). Now suppose that due to wear, one of the design parameters associated with that solution, changes (say  $x_1$ ). This will cause the performances to deteriorate (see the triangle in both panels). Now suppose that there is a way to actively change the remaining parameter  $x_2$  by actively controlling its value. If this is done properly, the performances might be improved to new performance vectors (designated in the figures by squares). The way the deteriorated performances are steered (controlled) as close as possible to the original location, has been termed in [11] as ‘control in objective space’. In [11] the control



**Fig. 2** Hypothetical two dimensional bi-objective design problem

problem has been defined as a regulative control problem, and a proportional controller has been used to update the optimal solution in time. Note, however, that since optimality is defined in objective space, the DS (or DDS) can be used to accomplish this task: The direction  $d_i$  is simply the difference of performance of the desired solution and the performance of the actual performance at the (deteriorated) point  $x_i$ , i.e.,  $d_i = F(x^*) - F(x_i)$ .

To illustrate the performance of the suggested control scheme, we choose the quadratic bi-objective problem

$$F : \mathbb{R}^{15} \rightarrow \mathbb{R}^2$$

$$F(x) = (\|x - a_1\|_2^2, \|x - a_2\|_2^2), \tag{36}$$

where  $a_1 = (1, \dots, 1)$  and  $a_2 = (-1, \dots, -1)$ . For the decision space we choose  $n = 15$  whereof three parameters deteriorate ( $x_1, x_2,$  and  $x_3$ ) and the rest can be controlled. The results for the handling of the DDS controller with the above problem are depicted in Figure 3 (a). The solid line represents the initial Pareto front. The circle is the performance of the initial design. The performance after deterioration occurs is marked with triangles. Since some design variables have deteriorated, the Pareto front has changes in time. The deteriorated Pareto front in every time step is marked with a dashed line. The final state of the DDS controlled performance is marked with a black square, while the trajectory is marked with smaller gray squares. Note that the trajectory is going along the Pareto optimal front, and stops when the error is minimal. Figure 3 (b) depicts the performance of the deteriorated product with and without the DDS controller. The uncontrolled performance is described with triangles and the controlled one with squares.

We note that this result has been obtained by using the classical DS, however, from this we conclude that the DDS might be an alternative choice for models where no gradient information is at hand. We leave this for future research.

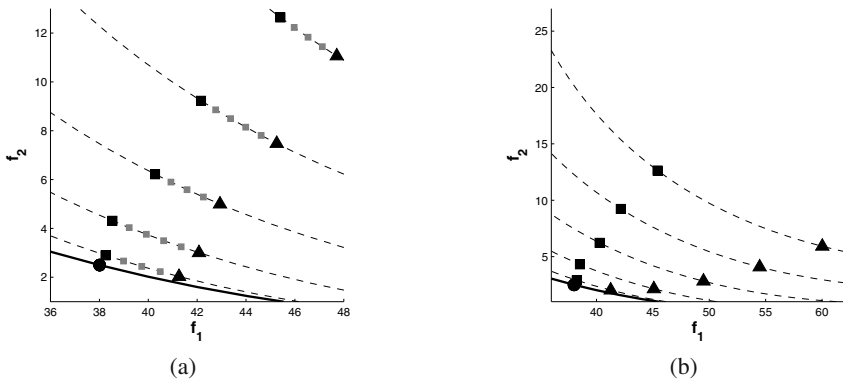


Fig. 3 Result of the DS approach to the control problem

## 6 Conclusions

In this paper, we have modified the Directed Search Method, a point-wise iterative search procedure that allows to steer the search into any direction given in objective space of a given MOP. The resulting algorithm, DDS, allows to perform similar iterations as its original, however, without using gradient information but by exploiting the neighborhood information in order to find a suitable search direction. The latter makes the new algorithm in particular interesting for set oriented search procedures such as MOEAs. Here, we have made a first attempt to demonstrate this by integrating the DDS into MOEA/D. Comparisons on some benchmark functions have shown the benefit of such a hybridization.

For future work, the development of more efficient memetic strategies as the one proposed in this paper is an interesting topic which will call for a more sophisticated interplay of local and global search. Also, the adaption of the DDS to higher dimensional problems seems to be very interesting. Note that the choice of the number of test points near a solution  $x_0$  that have to be chosen in order to find a search direction merely depends on the number of objectives involved in the MOP, and not on the dimension of the parameter space. Finally, we intend to utilize the DS/DDS in other applications, e.g., in the context of changing market demands as described in [2].

**Acknowledgements.** This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Program. The first author acknowledges support from IPN project no. 20121478, and the last author acknowledges support from CONACyT project no. 128554.

## References

1. Avigad, G., Eisenstadt, E.: Robustness of Multi-objective Optimal Solutions to Physical Deterioration through Active Control. In: Deb, K., Bhattacharya, A., Chakraborti, N., Chakraborty, P., Das, S., Dutta, J., Gupta, S.K., Jain, A., Aggarwal, V., Branke, J., Louis, S.J., Tan, K.C. (eds.) SEAL 2010. LNCS, vol. 6457, pp. 394–403. Springer, Heidelberg (2010)
2. Avigad, G., Eisenstadt, E., Schütze, O.: Handling changes of performance-requirements in multi objective problems. *Journal of Engineering Design* (to appear, 2012)
3. Bosman, P.A.N., de Jong, E.D.: Exploiting gradient information in numerical multi-objective evolutionary optimization. In: Beyer, H.-G., et al. (eds.) 2005 Genetic and Evolutionary Computation Conference (GECCO 2005), vol. 1, pp. 755–762. ACM Press, New York (2005)
4. Brown, M., Smith, R.E.: Directed multi-objective optimisation. *International Journal of Computers, Systems and Signals* 6(1), 3–17 (2005)
5. Coello Coello, C.A., Cruz Cortés, N.: Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines* 6(2), 163–190 (2005)
6. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edn. Springer, New York (2007)
7. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X

8. Ishibuchi, T.Y.H., Murata, T.: Balance between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), pp. 1301–1308. Morgan Kaufmann Publishers, San Francisco (2002)
9. Ishibuchi, H., Murata, T.: Multi-objective genetic local search algorithm. In: Proc. of 3rd IEEE Int. Conf. on Evolutionary Computation, Nagoya, Japan, pp. 119–124 (1996)
10. Jaskiewicz, A.: Do multiple-objective metaheuristics deliver on their promises? a computational experiment on the set-covering problem. *IEEE Transactions on Evolutionary Computation* 7(2), 133–143 (2003)
11. J.: D Knowles and D.W Corne. M-PAES: a memetic algorithm for multiobjective optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, New Jersey, pp. 325–332 (2000)
12. Köppen, M., Yoshida, K.: Many-Objective Particle Swarm Optimization by Gradual Leader Selection. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 323–331. Springer, Heidelberg (2007)
13. Lara, A., Coello Coello, C.A., Schütze, O.: A painless gradient-assisted multi-objective memetic mechanism for solving continuous bi-objective optimization problems. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE, IEEE Press (2010)
14. Lara, A., Sanchez, G., Coello Coello, C.A., Schütze, O.: HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 14(1), 112–132 (2010)
15. Mejia, E., Schütze, O.: A predictor corrector method for the computation of boundary points of a multi-objective optimization problem. In: International Conference on Electrical Engineering, Computing Science and Automati Control (CCE 2010), pp. 1–6 (2007)
16. Miettinen, K.M.: *Nonlinear Multiobjective Optimization*. Springer (1999)
17. Nocedal, J., Wright, S.: *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer (2006)
18. Schäffler, S., Schultz, R., Weinzierl, K.: A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications* 114(1), 209–222 (2002)
19. Schuetze, O., Equivel, X., Lara, A., Coello Coello, C.A.: Some comments on GD and IGD and relations to the Hausdorff distance. In: GECCO 2010: Proceedings of the 12th Annual Conference Comp. on Genetic and Evolutionary Computation, pp. 1971–1974. ACM, New York (2010)
20. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* (2012), doi:10.1109/TEVC.2011.2161872
21. Schütze, O., Lara, A., Coello Coello, C.A.: The directed search method for unconstrained multi-objective optimization problems. In: Proceedings of the EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation (2011)
22. Vasile, M.: A behavior-based meta-heuristic for robust global trajectory optimization. In: IEEE Congress on Evolutionary Computing, vol. 2, pp. 494–497 (2007)
23. Van Veldhuizen, D.A.: *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio (May 1999)
24. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. multi-objective evolutionary algorithm based on decomposition 11(6), 712–731 (2007)

25. Zhang, Q., Liu, W., Li, H.: The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 203–208. IEEE (2009)
26. Zhang, Q., Suganthan, P.N.: Final report on CEC09 MOEA competition. In: Congress on Evolutionary Computation, CEC 2009 (2009)
27. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. University of Essex, Technical Report (2008)
28. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4), 257–271 (1999)

**Part IV**  
**Combinatorial Optimization**



# A Hyperheuristic Approach for Guiding Enumeration in Constraint Solving

Broderick Crawford, Carlos Castro, Eric Monfroy, Ricardo Soto,  
Wenceslao Palma, and Fernando Paredes

**Abstract.** In this paper we design and evaluate a dynamic selection mechanism of enumeration strategies based on the information of the solving process. Unlike previous research works we focus in reacting on the fly, allowing an early replacement of bad-performance strategies without waiting the entire solution process or an exhaustive analysis of a given class of problems. Our approach uses a hyperheuristic approach that operates at a higher level of abstraction than the Constraint Satisfaction Problems solver. The hyperheuristic has no problem-specific knowledge. It manages a portfolio of enumeration strategies. At any given time the hyperheuristic must choose which enumeration strategy to call. The experimental results show the effectiveness of our approach where our combination of strategies outperforms the use of individual strategies.

## 1 Introduction

Constraint Programming (CP) is a powerful software technology devoted to the efficient resolution of constraint-based problems. Currently, CP is largely used in

---

Broderick Crawford · Ricardo Soto  
Pontificia Universidad Católica de Valparaíso, Chile  
e-mail: [broderick.crawford@ucv.cl](mailto:broderick.crawford@ucv.cl)

Carlos Castro  
Universidad Técnica Federico Santa María, Chile  
e-mail: [carlos.castro@inf.utfsm.cl](mailto:carlos.castro@inf.utfsm.cl), [ricardo.soto@ucv.cl](mailto:ricardo.soto@ucv.cl)

Eric Monfroy  
CNRS, LINA, Université de Nantes, France  
e-mail: [ericmonfroy@gmail.com](mailto:ericmonfroy@gmail.com)

Wenceslao Palma  
Pontificia Universidad Católica de Valparaíso, Chile  
e-mail: [wenceslao.palma@ucv.cl](mailto:wenceslao.palma@ucv.cl)

Fernando Paredes  
Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile  
e-mail: [fernando.paredes@udp.cl](mailto:fernando.paredes@udp.cl)

different application domains, for instance, in computer graphics, natural language processing, database systems, electrical engineering, and even for sequencing the DNA in molecular biology.

In CP, the selection of an enumeration strategy is crucial for the performance of the resolution process, where a correct selection can dramatically reduce the computational cost of finding a solution. However, it is well-known that deciding a priori the correct heuristic is quite difficult, as the effects of the strategy can be unpredictable. In recent years, different efforts have been done to determine good strategies based on the information generated through the resolution process. However, deciding what information must be measured and how to redirect the search is an ongoing research direction.

There exist various studies about enumeration strategies [2, 5], some centered in defining general criteria, e.g., the smallest domain for variable selection, and its minimum, maximum, or a random value for value selection. As opposed to this idea, some research works have proposed strategies for a given class of problems, e.g., for job shop scheduling [14], as well as for configuration design [6]. We can also find research focused on determining the best strategy based on some criterion [15], i.e., the selection heuristic is determined only once before starting the resolution process, remaining unchangeable during the whole process. However, deciding a priori the correct heuristics is quite hard, as the effects of the strategy can be unpredictable. In recent years, multiple efforts have been done to determine good strategies based on the information generated through the resolution process.

Cowling et al. first introduced the term hyperheuristic [7]. In their words, a hyperheuristic is an “approach that operates at a higher level of abstraction than metaheuristics and manages the choice of which low-level heuristic method should be applied at any given time, depending upon the characteristics of the region of the solution space currently under exploration.” This means that the hyperheuristic itself does not search for a better solution to the problem, instead, it selects at each step of the solution process the most promising simple low-level heuristic (or combination of heuristics) which is potentially able to improve the solution. On the other hand, if there is no improvement, i.e., a locally optimal solution is found, the hyperheuristic diversifies the search to another area of the solution space by selecting appropriate heuristics from the given set.

Low-level heuristics usually represent the simple local search neighborhoods or the rules used by human experts for constructing solutions. However, it is also possible that more complex heuristics such as metaheuristics can be considered at a lower level. All domain-specific information is concentrated in the set of low-level heuristics and the objective function. Hyperheuristics do not require knowledge of how each low-level heuristic works or the contents of the objective function of the problem (other than the value returned). It only needs to know the direction of the optimization process (maximizing or minimizing) and analyzes the value of one

or more objective functions and, sometimes, the amount of CPU time required to perturb the solution, which are returned by the low-level heuristic after its call.

Considering the aforementioned concerns, we are interested in making good choices for enumeration, i.e., selection of a variable and a value. However, unlike previous research works we focus our research in reacting on the fly, allowing an early replacement of bad-performance strategies without waiting the entire solution process or an exhaustive analysis of a given class of problems. Regarding this issue, we use Autonomous Search (AS) principles [9] where a system should be able to change its internal components when exposed to changing external forces and opportunities, in order to obtain better results. More precisely, we use a hyperheuristic approach that operates at a higher level of abstraction than the CSP solver. The hyperheuristic has no problem-specific knowledge. It manages a portfolio of enumeration strategies. At any given time the hyperheuristic must choose which enumeration strategy to call. Our approach exploits search process features to dynamically adapt a CP solver changing the enumeration strategy in use when another strategy looks more promising in order to solve the CSP at hand.

The contributions of this paper are the following:

- We propose a hyperheuristic approach to decide which enumeration strategy to apply at each decision step during the search.
- We define a choice function which adaptively ranks the enumeration strategies allowing our approach be able to detect a poor search performance and, as a result, replace the enumeration strategy with a better one. The problem of determining the best set of parameters of the choice function is tackled using a Genetic Algorithm (GA). Thus,
- We identify and measure a set of indicators which allows the classification of the execution process state such that we are able to determine if the current strategy exhibits a poor performance. We perform a correlation analysis in order to detect indicators that are not useful because they behave as another indicator.

The rest of this paper is organized as follows. In Section 2 we present our approach to meet the challenge of dynamic selecting enumeration strategies for solving CSPs. Section 3 presents the benchmark problems and the experiments settings. We then present the experiments and the analysis of results. Finally, in Section 5 we conclude and highlight future directions of research.

## 2 A Hyperheuristic Approach for Dynamic Selection of Enumeration Strategies

One of the main goals of using hyperheuristics is the aim of achieving robustness in terms of good-enough, cheap-enough, soon-enough solutions across a wide range of problems and domains. A hyperheuristic itself is a heuristic that drives the search process at a high level by controlling a set of low level heuristics. These low level heuristics can be perturbative or constructive. However it would be difficult

to achieve its main goals without a certain level of learning ability, which might be needed in order for the hyperheuristic to solve different problems in an effective manner.

In this section we present a hyperheuristic approach which decides which enumeration strategy to apply at each decision step during the search. Then, we present the details of the proposal including the indicators used to measure the solving process, a Choice Function that determines the performance of a given enumeration strategy in a given amount of time through a set of indicators and control parameters. Finally, we present a GA used to tune the parameters of the choice function.

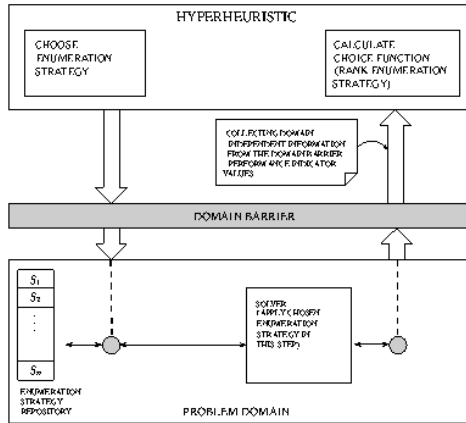


Fig. 1 Hyperheuristic framework based on a choice function

## 2.1 The Hyperheuristic Approach

Hyperheuristic systems employ one of two approaches in deciding on which constructive heuristic to apply next. The first either identifies or adapts an existing heuristic to be applied at each stage of the construction process, while the second approach optimizes a search space of heuristic combinations, i.e., list of low-level heuristics. The study presented in this work focuses on the second approach taken by hyperheuristic systems, i.e., the generation of combinations of low-level heuristics. To allow the hyperheuristic to operate, we define a Choice Function which adaptively ranks the enumeration strategies [8]. The Choice Function provides guidance to the hyperheuristic (see Figure 1) by indicating which enumeration strategy should be applied next based upon the information of the search process (it should be captured through some indicators). The Choice Function is defined as a weighted sum of indicators (a linear combination) expressing the recent improvement produced by the enumeration strategy called.

## 2.2 Measuring the Solving Process: Indicators

An indicator provides information that will enable us to know the state of progress in the solution process for a problem. Indicators can be divided into *base indicators* and *calculated indicators* (from the base indicators), see Table 5 and 6 respectively.

### 2.2.1 Indicators for the Search-Tree Process Cost

The cost measures allow us to decide when one strategy performs better than another [3]. While the comparison with other techniques may be done once the resolution of the problem has ended, some of these indicators could be used in intermediate stages of the process to assess the performance of resolution process at a specific time. Example of this kind of indicators are:

- Number of backtracks (Backtracks) [3, 4, 13, 15]: Counts the number of times the resolution process goes back from a variable  $x_j$  to its predecessor  $x_i$  after having proved that none of the extensions of  $x_j$  can be extended to a solution. In terms of the search tree, it counts the number of times the search goes up in the tree from a node  $u$  to its predecessor after having exhausted the last child of  $u$ . The condition for a node  $u$  to be counted as a backtrack node is that it has some children which have been visited, and that none of them was successful.
- Number of nodes (Nodes) [3]: Counts the number of visited nodes.
- Runtime/CPU time [4, 13]: Measures the time required in solution process of a problem.

Collecting the indicators and using all of them in the choice function is a costly task (it is a combinatorial problem by itself). We thus use correlation analysis to detect indicators that are not useful because they behave as another indicator. A correlation analysis was performed to detect pairs or sets of highly related indicators, and thus equivalent. We use the Pearson Correlation Coefficient, which is sensitive to a linear relationship between two variables. The criteria used to detect pairs of indicators highly related was an absolute value of correlation coefficient at least of 0,9. We identified the indicators highly related (see Table 1), then we will use only one of them (one per row) in our Choice Functions. Thus, for example, we will use  $VF$  (the number of fixed variables) and not  $d$  (the current depth in the search tree) since they are highly related, and thus redundant.

**Table 1** Indicators highly related (group by row)

Positive Linear Relationship				Negative Linear Relationship		
1	B	N	TAVFES	TSB	PTAVFES	DB
2	VF	d				
3	VFPS	PVFP				
4	d_pr	Thrash				

### 2.3 Choice Function

As mentioned above, our hyperheuristic is based upon a choice function, which adaptively ranks the enumeration strategies. The choice function value of each enumeration strategy is determined based on information with regards to performance indicators. The choice function attempts to capture the correspondence between the historical performance of each enumeration strategy and the decision point currently being investigated. Here, a decision point or step is performed every time the solver is invoked to fix a variable by enumeration.

The choice function is used to rank and choose between different enumeration strategies at each step. For any enumeration strategy  $S_j$ , the choice function  $f$  in step  $n$  for  $S_j$  is defined by Equation 1 where  $l$  is the number of indicators considered and  $\alpha$  is a parameter to control the relevance of the indicator within the choice function.

$$f_n(S_j) = \sum_{i=1}^l \alpha_i f_{i_n}(S_j) \quad (1)$$

Additionally, to control the relevance of an indicator  $i$  for a strategy  $S_j$  in a period of time, we use a popular statistical technique for producing smoothed time series called exponential smoothing. The idea is to associate greater importance to recent performance by exponentially decreasing weights to older observations. The exponential smoothing is applied to the computation of  $f_{i_n}(S_j)$ , which is defined by equations 2 and 3 where  $v_{i_1}$  is the value of the indicator  $i$  for the strategy  $S_j$  in time 1,  $n$  is a given step of the process,  $\beta$  is the smoothing factor, and  $0 < \beta < 1$ .

$$f_{i_1}(S_j) = v_{i_1} \quad (2)$$

$$f_{i_n}(S_j) = v_{i_n} + \beta_i f_{i_{n-1}}(S_j) \quad (3)$$

Then the exponentially smoothed moving average for step  $n$  is given by

$$f_{i_n}(S_j) = v_{i_n} + \beta_i v_{i_{n-1}} + \beta_i^2 v_{i_{n-2}} + \beta_i^3 v_{i_{n-3}} + \dots \quad (4)$$

Let us note that the speed at which the older observations are smoothed (dampened) depends on  $\beta$ . When  $\beta$  is close to 0, dampening is quick, and when it is close to 1, dampening is slow.

### 2.4 Choice Function Tuning with a Multilevel Structure

The search for the best tuning of parameters  $\alpha_i$  of the CSP solver (based on the Choice Function) may be formulated as an optimization problem. Hence, this meta-optimization approach may be performed by a metaheuristic. This approach is composed of two levels [16]: the meta-level and the base level. At the meta-level,

a metaheuristic operates on solutions representing the parameters of the metaheuristic to optimize. A solution  $X$  at the meta-level will represent all the parameters we want to optimize. At the meta-level, the objective function  $f_m$  associated with a solution  $X$  is a performance indicator of the metaheuristic using the parameters specified by the solution  $X$ . Hence, to each solution  $X$  of the meta-level will correspond an independent metaheuristic in the base level. The metaheuristic of the base level operates on solutions that encode solutions of the original problem. The objective function  $f_b$  used by the metaheuristics of the base level is associated with the target problem. The study presented in this work at the meta-level employs a Genetic Algorithm to search the heuristic space of choice functions of the base level (see Figure 2).

Then, in order to determine the most appropriate set of parameters  $\alpha_i$  for the choice function we use a multilevel approach. The parameters are fine-tuned by a GA which trains the choice function carrying out a sampling phase. Sampling occurs during an initial information gathering phase where the search is run repeatedly until a fix cutoff (i.e., until a fixed number of variables instantiated, number of visited nodes, number of backtracks or number of steps). After sampling, the problem is solved with the most promising set of parameter values for the choice function.

The upper-level GA simulates, evaluates and evolves different combinations of parameters, relieving the task of manual parameterization. Each member of the population encodes the parameters of a choice function, then these individuals are used in order to create a choice function instance. Each choice function instantiated (i.e. each chromosome) is evaluated in a sampling phase trying to solve partially the problem at hand. An indicator of process performance is used as a fitness value for the chromosome, (number of visited nodes, number of backtracks or number of steps). After each chromosome of the population is evaluated, selection, crossover and mutation are used to breed a new population of choice functions. As noted above, the multilevel approach is used to tune the choice function, the resulting choice function is applied to solve the entire CSP problem.

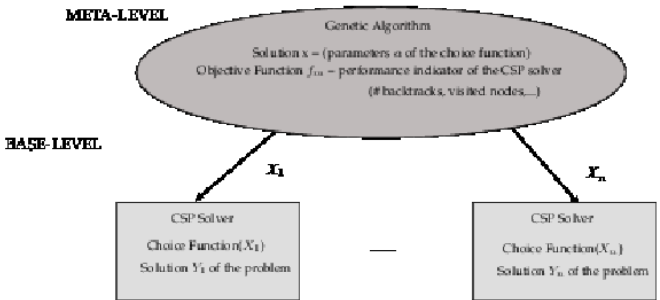


Fig. 2 Multilevel parameterization

### 3 Experimental Evaluation

In this Section, we provide a performance evaluation of our approach. We have implemented a solver using ECL<sup>i</sup>PS<sup>e</sup> Constraint Programming System version 5.10 and java with NetBeans IDE 6.7.1. Tests have been performed on a 2.33GHZ Intel Core2 Duo with 2GB RAM running Windows XP. We describe the benchmark problems used to test our approach. We then evaluate our proposal using the benchmark problems and we analyze the results. The problems used to test our approach were the following: N-queens (NQ), Magic square (MS) and Sudoku.

The variable selection heuristics used in the experiments are:

- *input\_order*: the first entry in the list is selected.
- *anti\_first\_fail*: the entry with the largest domain size is selected.
- *first\_fail*: the entry with the smallest domain size is selected (also known as minimum remaining values).
- *occurrence*: the entry with the largest number of attached constraints is selected.

The value selection heuristics used are:

- *indomain(Var,min)*: it starts with the smallest element and upon backtracking tries successive elements until the entire domain has been explored
- *indomain(Var,max)*: it starts the enumeration from the largest value downwards

The combination of variable and value selection heuristics generates eight enumeration strategies as shown in Table 2.

**Table 2** Enumeration strategies used

$S_1 = \text{input\_order} + \text{indomain}(\text{Var}, \text{min})$
$S_2 = \text{anti\_first\_fail} + \text{indomain}(\text{Var}, \text{min})$
$S_3 = \text{first\_fail} + \text{indomain}(\text{Var}, \text{min})$
$S_4 = \text{occurrence} + \text{indomain}(\text{Var}, \text{min})$
$S_5 = \text{input\_order} + \text{indomain}(\text{Var}, \text{max})$
$S_6 = \text{anti\_first\_fail} + \text{indomain}(\text{Var}, \text{max})$
$S_7 = \text{first\_fail} + \text{indomain}(\text{Var}, \text{max})$
$S_8 = \text{occurrence} + \text{indomain}(\text{Var}, \text{max})$

Considering the correlation analysis of the indicators we considered the following combinations of them in order to test some different choice functions (see Table 3).

The genetic algorithm used to obtain the best weights ( $\alpha$ 's) of the choice functions was implemented using the parameters showed in Table 4. The GA was implemented using the Java Genetic Algorithm Package<sup>1</sup> (JGAP) version 3.5. The basic behavior of a GA implemented using JGAP contains three main steps: a setup phase, the creation of the initial population and the evolution of the population.

As mentioned above, our main goal is to determine if the current strategy exhibits a poor performance and whether it is necessary to replace it with a better one. We are thus concerned to apply a combination of strategies rather than only one. Our

<sup>1</sup> <http://jgap.sourceforge.net>



**Table 3** Choice functions used in the experiments

$CF_1: \alpha_1 B + \alpha_2 VF + \alpha_3 VFPS + \alpha_4 Thrash$	$CF_{13}: \alpha_1 N + \alpha_2 d + \alpha_3 VFPS + \alpha_4 Thrash$
$CF_2: \alpha_1 B + \alpha_2 VF + \alpha_3 VFPS + \alpha_4 d_{pr}$	$CF_{14}: \alpha_1 N + \alpha_2 d + \alpha_3 VFPS + \alpha_4 d_{pr}$
$CF_3: \alpha_1 B + \alpha_2 VF + \alpha_3 PVFPS + \alpha_4 Thrash$	$CF_{15}: \alpha_1 N + \alpha_2 d + \alpha_3 PVFPS + \alpha_4 Thrash$
$CF_4: \alpha_1 B + \alpha_2 VF + \alpha_3 PVFPS + \alpha_4 d_{pr}$	$CF_{16}: \alpha_1 N + \alpha_2 d + \alpha_3 PVFPS + \alpha_4 d_{pr}$
$CF_5: \alpha_1 B + \alpha_2 d + \alpha_3 VFPS + \alpha_4 Thrash$	$CF_{17}: \alpha_1 B + \alpha_2 VU + \alpha_3 VFPS + \alpha_4 Thrash$
$CF_6: \alpha_1 B + \alpha_2 d + \alpha_3 VFPS + \alpha_4 d_{pr}$	$CF_{18}: \alpha_1 B + \alpha_2 VU + \alpha_3 VFPS + \alpha_4 d_{pr}$
$CF_7: \alpha_1 B + \alpha_2 d + \alpha_3 PVFPS + \alpha_4 Thrash$	$CF_{19}: \alpha_1 B + \alpha_2 VU + \alpha_3 PVFPS + \alpha_4 Thrash$
$CF_8: \alpha_1 B + \alpha_2 d + \alpha_3 PVFPS + \alpha_4 d_{pr}$	$CF_{20}: \alpha_1 B + \alpha_2 VU + \alpha_3 PVFPS + \alpha_4 d_{pr}$
$CF_9: \alpha_1 N + \alpha_2 VF + \alpha_3 VFPS + \alpha_4 Thrash$	$CF_{21}: \alpha_1 N + \alpha_2 VU + \alpha_3 VFPS + \alpha_4 Thrash$
$CF_{10}: \alpha_1 N + \alpha_2 VF + \alpha_3 VFPS + \alpha_4 d_{pr}$	$CF_{22}: \alpha_1 N + \alpha_2 VU + \alpha_3 VFPS + \alpha_4 d_{pr}$
$CF_{11}: \alpha_1 N + \alpha_2 VF + \alpha_3 PVFPS + \alpha_4 Thrash$	$CF_{23}: \alpha_1 N + \alpha_2 VU + \alpha_3 PVFPS + \alpha_4 Thrash$
$CF_{12}: \alpha_1 N + \alpha_2 VF + \alpha_3 PVFPS + \alpha_4 d_{pr}$	$CF_{24}: \alpha_1 N + \alpha_2 VU + \alpha_3 PVFPS + \alpha_4 d_{pr}$

**Table 4** GA parameters

Parameter	Value
Number of generations	20
Population size	30
Crossover type	Uniform crossover
Crossover rate	0.4
Mask probability	0.5
Mutation rate	0.025
Selector tournament size	3
Tournament selector parameter ( $p$ )	0.75
Fitness	number of visited nodes

solver uses a single strategy at a time, but when a strategy is judged (by the choice function) to have a poor behaviour, it is replaced by another one in order to explore differently the search tree.

For comparison with our choice function approach (using the portfolio of 8 enumeration strategies), we considered 8 enumeration strategies used solely (F+ID, AMRV + ID, MRV + ID, O + ID, F + IDM, AMRV + IDM, MRV + IDM, and O + IDM), and a random selection. In the experiments with Random and Choice Function approaches we show **min** (the minimum value), **avg** (the average obtained of the experiments where a solution was found before the timeout of 65535 steps), **sd** (the standard deviation), **max** (the maximum value) and [%] (the percentage of the experiments performed before the timeout is reached).

### 3.1 Results

For each measurement, the results were obtained over 30 experiments. Tables 7, 8 and 9 present the results measured in terms of number of backtracks. More precisely, Table 7 shows the results related to the N-Queens problem. Tables 8 and 9 show the results related to the problems Magic square and Sudoku. Table 10 presents the ranking of the competing approaches w.r.t N-Queens, Magic square and Sudoku respectively.

Table 5 Base Indicators

Code	Name (Base Indicators)	Description	Calculate	References
Step	Number of Measurements	Every time the choice function is used	Step ++	
TV	Total Number of Variables	Total Number of Variables in the Problem. It is an indicator in relation with the characteristic of the problem, rather than the solving process. Its value is constant during the process	$\text{length}(\text{AllVars}, N)$	
VU	Variables Unfixed	Number of Variables Unfixed	$\text{var.count}(\text{Rest}, \text{VU})$	Conceptually corresponds to $\text{length}(\text{Rest}, N) - \text{VU}$ . Where "VU" is the number of variables with domain size equal to 1 in the Variables Unfixed set. Note that "VU" is equal to VFPS (see Table 6).
SB	Shallow Backtracks	When we try to assign a value to the current variable, the constraint propagation process is automatically triggered. If it yields a failure, the next value is tried.		[1]
SS	Search Space	Current Search Space	$\Pi(\text{Dom}_i)$	[13]
SS-pr	Previous Search Space	Previous Search Space	$\Pi(\text{Dom}_{i-1})$	[13]
B	Backtracks	If the Current Variable causes a Dead-End (Propagation that produces an Empty Domain) then the Algorithm Backtracks to the Previously Instantiated Variable	Every Time you Try to Perform a B. A counter is set to increase by one	[3, 4, 10, 11, 13, 15]
N	Nodes	Number of Nodes Visited	A counter is set to increase by one every time you visit a node	[3]

Table 6 Calculated Indicators

Code	Name (Calculated Indicators)	Description	Calculate	References
VF	Variables Fixed	Number of Variables Fixed by Enumeration and Propagation	$(TV - VU)$	[5, 10]
VFES	Variables Fixed by Enumeration in each Step	Number of Variables Fixed by Enumeration in each Step. It makes 1 sense when the number of steps is greater than 1	1	[5, 13]
TAVFES	Total Accumulated of Variables Fixed by Enumeration in each Step	Number of Total Accumulated Variables Fixed by Enumeration in each Step	$\Sigma(VFES)$	[5, 13]
VFPS	Variables Fixed by Propagation in each Step	Number of Variables Fixed by Propagation in each Step. Note that $length(Res, N) - VU$		[13]
TAVFPS	Total Accumulated of Variables Fixed by Propagation in each Step	VFPS is equal to "Vt1" (see Table 5)	$\Sigma(VFPS)$	[13]
TSB	Total Shallow Backtracks	Accumulated Number of Shallow Backtracks	$\Sigma(SB)$	[1]
$d_{-pr}$	Previous Depth	Previous Depth of the current node in the Search Tree		[13]
d	Depth	Current Depth of the current node in the Search Tree		[5, 13]
dmax	Maximum Depth	Current Maximum Depth Reached in the Search Tree	$dmax$ is $d$ , if $d \geq dmax_{-pr}$	[13]
$dmax_{-pr}$	Previous Maximum Depth	Previous Maximum Depth Reached in the Search Tree	$MAX(dmax)_{n-1}$	[13]
In1	Current Maximum Depth - Previous Maximum Depth	Represents a Variation of the Maximum Depth.	$(dmax - dmax_{-pr})$	[13]
In2	Current Depth - Previous Depth	Positive means, that the Current Node is Deeper than the one Explored at the Previous Step.	$(d_t - d_{t-1})$	[13]
In3	Reduction Search Space	If Positive, The Current Search Space is Smaller than the one at Previous Snapshot.	$((SS_{t-1} - SS_t) / SS_{t-1}) * 100$	[13]
PVFES	Percentage of Variables Fixed by Enumeration in each Step	It makes sense when the number of steps is greater than 1	$((VFES / TV) * 100)$	[5]
PTAVFES	Percentage of Total Accumulated of Variables Fixed by Enumeration in each Step		$((TAVFES / TV) * 100)$	[5]
PVFPS	Percentage of Variables Fixed by Propagation in each Step		$((VFPS / TV) * 100)$	[5]
PTAVFPS	Percentage of Total Number of Variables Fixed by Propagation in each Step		$((PTAVFPS / TV) * 100)$	[5]
Thrash	Thrashing	The Solving Process alternates Enumerations and Backtracks on a few Number of Variables without Succeeding in having a Strong Orientation.	$(d_{t-1} - VFPS_{t-1})$	[5]
DB	Depth Back	If a backtrack is performed. Count how many levels backward in the Depth	If $D - d_{pr} = 0$ , then add 1. If $D < d_{pr}$ , then increases in $(d_{pr} - D) + 1$ . In another case maintain constant	[5]

Table 7 Number of Backtracks solving N-Queens problem with different strategies

Strategy	NQ n=8	NQ n=9	NQ n=10	NQ n=11	NQ n=12
$S_1$	10	2	6	2	15
$S_2$	11	2	12	39	11
$S_3$	10	2	4	5	16
$S_4$	10	2	6	2	15
$S_5$	10	2	6	2	15
$S_6$	11	2	12	39	11
$S_7$	10	2	4	5	16
$S_8$	10	2	6	2	15
Strategy	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]
Random	(4) 7.5 $\pm$ 2.2 (12) [100]	(1) 5.3 $\pm$ 3.6 (11) [100]	(2) 9.5 $\pm$ 5.3 (21) [100]	(0) 15.5 $\pm$ 8.6 (33) [100]	(2) 16.4 $\pm$ 11.7 (39) [100]
$CF_1$	(4) 5.7 $\pm$ 2.2 (10) [100]	(1) 4.5 $\pm$ 3.6 (11) [100]	(4) 7.1 $\pm$ 2.1 (11) [100]	(0) 5.1 $\pm$ 6.4 (21) [100]	(1) 14.9 $\pm$ 9.8 (48) [100]
$CF_2$	(4) 6.3 $\pm$ 2.5 (10) [100]	(1) 3.8 $\pm$ 3.4 (10) [100]	(4) 6.4 $\pm$ 1.5 (9) [100]	(0) 5.1 $\pm$ 8.3 (38) [100]	(0) 16.5 $\pm$ 13.4 (46) [100]
$CF_3$	(4) 6.4 $\pm$ 2.2 (10) [100]	(1) 4.4 $\pm$ 3.2 (10) [100]	(2) 6.4 $\pm$ 1.9 (10) [100]	(0) 5.9 $\pm$ 7.5 (27) [100]	(0) 11.5 $\pm$ 10.4 (48) [100]
$CF_4$	(4) 6.4 $\pm$ 2.2 (10) [100]	(1) 4.4 $\pm$ 3.1 (10) [100]	(4) 7 $\pm$ 1.6 (10) [100]	(0) 4 $\pm$ 5.1 (20) [100]	(0) 13.8 $\pm$ 7.9 (25) [100]
$CF_5$	(4) 5.6 $\pm$ 2.2 (10) [100]	(1) 4.2 $\pm$ 3.5 (10) [100]	(4) 6.5 $\pm$ 1.5 (10) [100]	(0) 4.7 $\pm$ 7.8 (34) [100]	(0) 14.2 $\pm$ 10.6 (47) [100]
$CF_6$	(4) 5.8 $\pm$ 2.1 (10) [100]	(1) 4.8 $\pm$ 3.9 (10) [100]	(4) 6.4 $\pm$ 1.4 (11) [100]	(0) 4.1 $\pm$ 5 (16) [100]	(2) 13.9 $\pm$ 8 (30) [100]
$CF_7$	(4) 6.2 $\pm$ 2.2 (10) [100]	(1) 3.9 $\pm$ 3.2 (10) [100]	(5) 6.6 $\pm$ 1.3 (10) [100]	(0) 4.4 $\pm$ 6.8 (35) [100]	(0) 12 $\pm$ 8.8 (29) [100]
$CF_8$	(4) 5.7 $\pm$ 2.1 (10) [100]	(1) 4.4 $\pm$ 3.7 (10) [100]	(4) 6.8 $\pm$ 1.4 (10) [100]	(0) 6.5 $\pm$ 9.9 (37) [100]	(3) 16.7 $\pm$ 13 (50) [100]
$CF_9$	(4) 6 $\pm$ 2 (10) [100]	(1) 4.5 $\pm$ 3.5 (11) [100]	(5) 7.3 $\pm$ 2 (12) [100]	(0) 5.2 $\pm$ 6.2 (19) [100]	(0) 9.9 $\pm$ 8.8 (31) [100]
$CF_{10}$	(4) 5.1 $\pm$ 1.7 (10) [100]	(1) 4.3 $\pm$ 3.5 (10) [100]	(4) 6.6 $\pm$ 1.6 (12) [100]	(0) 5.4 $\pm$ 7.5 (31) [100]	(1) 13.6 $\pm$ 9.7 (45) [100]
$CF_{11}$	(4) 6.2 $\pm$ 2.3 (10) [100]	(1) 4.2 $\pm$ 3.3 (10) [100]	(4) 6.7 $\pm$ 1.2 (10) [100]	(0) 6.6 $\pm$ 9 (32) [100]	(2) 12.4 $\pm$ 8.4 (39) [100]
$CF_{12}$	(4) 6.3 $\pm$ 2.1 (10) [100]	(1) 4.2 $\pm$ 3.4 (11) [100]	(4) 6.9 $\pm$ 1.9 (12) [100]	(0) 3.2 $\pm$ 4.9 (20) [100]	(0) 14.2 $\pm$ 10.8 (45) [100]
$CF_{13}$	(4) 6.2 $\pm$ 2.2 (10) [100]	(1) 4.5 $\pm$ 3.6 (11) [100]	(4) 6.6 $\pm$ 1.6 (10) [100]	(0) 4.9 $\pm$ 6 (17) [100]	(0) 15.2 $\pm$ 10.3 (44) [100]
$CF_{14}$	(4) 5.7 $\pm$ 2.1 (10) [100]	(1) 5.1 $\pm$ 3.7 (11) [100]	(5) 7.2 $\pm$ 1.4 (11) [100]	(0) 6.7 $\pm$ 8.7 (34) [100]	(0) 13.5 $\pm$ 10.8 (47) [100]
$CF_{15}$	(4) 5.9 $\pm$ 1.9 (10) [100]	(1) 4.3 $\pm$ 3.6 (10) [100]	(4) 6.9 $\pm$ 1.6 (11) [100]	(0) 4.7 $\pm$ 6.1 (21) [100]	(0) 9.5 $\pm$ 6.6 (21) [100]
$CF_{16}$	(4) 6.5 $\pm$ 2.3 (10) [100]	(1) 4.3 $\pm$ 3.4 (10) [100]	(4) 6.8 $\pm$ 1.6 (10) [100]	(0) 2.9 $\pm$ 4.4 (16) [100]	(1) 13.8 $\pm$ 7.9 (30) [100]
$CF_{17}$	(4) 6.8 $\pm$ 2.4 (10) [100]	(1) 4.1 $\pm$ 3.6 (10) [100]	(2) 6.4 $\pm$ 2.1 (13) [100]	(0) 5.4 $\pm$ 8.5 (41) [100]	(1) 15.5 $\pm$ 9.9 (37) [100]
$CF_{18}$	(4) 6.7 $\pm$ 1.9 (10) [100]	(1) 3.5 $\pm$ 3.4 (11) [100]	(4) 6.5 $\pm$ 1.7 (11) [100]	(0) 8 $\pm$ 8.9 (32) [100]	(1) 13.9 $\pm$ 12.5 (45) [100]
$CF_{19}$	(4) 6.4 $\pm$ 1.9 (10) [100]	(1) 3.3 $\pm$ 3 (11) [100]	(4) 6.9 $\pm$ 2.4 (16) [100]	(1) 6.2 $\pm$ 7.8 (34) [100]	(1) 12.7 $\pm$ 10 (47) [100]
$CF_{20}$	(4) 7.1 $\pm$ 2.5 (10) [100]	(1) 5.1 $\pm$ 3.8 (11) [100]	(2) 6.2 $\pm$ 2 (10) [100]	(0) 7.9 $\pm$ 9.6 (35) [100]	(0) 15.8 $\pm$ 10.1 (50) [100]
$CF_{21}$	(4) 5.9 $\pm$ 2.1 (10) [100]	(1) 4.5 $\pm$ 3.4 (10) [100]	(4) 6.8 $\pm$ 1.7 (11) [100]	(1) 6.3 $\pm$ 10.1 (35) [100]	(0) 15.4 $\pm$ 11.3 (45) [100]
$CF_{22}$	(4) 6 $\pm$ 2.1 (10) [100]	(1) 3.9 $\pm$ 3.5 (10) [100]	(4) 6.7 $\pm$ 1.4 (10) [100]	(0) 5.1 $\pm$ 6.7 (31) [100]	(0) 14 $\pm$ 8.1 (27) [100]
$CF_{23}$	(4) 6.4 $\pm$ 2.4 (10) [100]	(1) 3.6 $\pm$ 3.2 (10) [100]	(4) 6.7 $\pm$ 2.1 (13) [100]	(0) 2.8 $\pm$ 4.1 (16) [100]	(1) 14.6 $\pm$ 7.5 (36) [100]
$CF_{24}$	(4) 6 $\pm$ 2 (10) [100]	(1) 4.5 $\pm$ 3.6 (11) [100]	(2) 6.7 $\pm$ 2.6 (11) [100]	(0) 5.1 $\pm$ 6.6 (28) [100]	(0) 14.6 $\pm$ 11.5 (50) [100]

**Table 8** Number of Backtracks solving Magic Square problem with different strategies

Strategy	MS n=3	MS n=4	MS n=5
$S_1$	0	12	910
$S_2$	4	1191	>46675
$S_3$	0	3	185
$S_4$	0	10	5231
$S_5$	1	51	>47748
$S_6$	0	42	>44157
$S_7$	1	97	>47935
$S_8$	1	29	>39008
	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]
Random	(0) 0,7 $\pm$ 1,2 (4) [100]	(0) 31,1 $\pm$ 43,9 (184) [100]	(1) 3831 $\pm$ 8402,7 (37986) [66,7]
$CF_1$	(0) 0,7 $\pm$ 1 (4) [100]	(0) 51,8 $\pm$ 67,3 (181) [100]	(0) 1615,5 $\pm$ 4447,1 (19640) [70]
$CF_2$	(0) 0,5 $\pm$ 0,8 (4) [100]	(0) 51,9 $\pm$ 74,8 (182) [100]	(0) 976,9 $\pm$ 3246,3 (13113) [53,3]
$CF_3$	(0) 0,7 $\pm$ 1,2 (4) [100]	(0) 38,3 $\pm$ 50,3 (181) [100]	(0) 1502,5 $\pm$ 4562,7 (19633) [80]
$CF_4$	(0) 1,1 $\pm$ 1,5 (4) [100]	(0) 38,4 $\pm$ 50,8 (181) [100]	(0) 734,9 $\pm$ 2683,9 (12906) [76,7]
$CF_5$	(0) 0,5 $\pm$ 0,8 (4) [100]	(0) 45,3 $\pm$ 62,4 (181) [100]	(0) 892,6 $\pm$ 1689,4 (6346) [80]
$CF_6$	(0) 0,8 $\pm$ 1 (4) [100]	(0) 54,6 $\pm$ 58,2 (193) [100]	(0) 586,4 $\pm$ 1424,8 (6656) [83,3]
$CF_7$	(0) 0,8 $\pm$ 1,2 (4) [100]	(0) 69,9 $\pm$ 74,4 (181) [100]	(3) 1716,4 $\pm$ 3276,2 (12918) [76,7]
$CF_8$	(0) 0,8 $\pm$ 1,2 (4) [100]	(0) 45,2 $\pm$ 65,4 (181) [100]	(0) 219,9 $\pm$ 346,4 (1441) [70]
$CF_9$	(0) 0,9 $\pm$ 1,5 (4) [100]	(0) 35,7 $\pm$ 46,8 (149) [100]	(0) 683,1 $\pm$ 2464 (12906) [90]
$CF_{10}$	(0) 0,5 $\pm$ 0,8 (4) [100]	(0) 43,9 $\pm$ 52,8 (181) [100]	(0) 1047,6 $\pm$ 2846,9 (12906) [76,7]
$CF_{11}$	(0) 0,9 $\pm$ 1,3 (4) [100]	(0) 49,3 $\pm$ 62,2 (181) [100]	(4) 1233,3 $\pm$ 3421,4 (14823) [63,3]
$CF_{12}$	(0) 0,4 $\pm$ 0,8 (4) [100]	(0) 34,3 $\pm$ 52,6 (181) [100]	(0) 2253,5 $\pm$ 5095,7 (19632) [70]
$CF_{13}$	(0) 1,2 $\pm$ 1,6 (4) [100]	(0) 51,3 $\pm$ 59,7 (181) [100]	(15) 1149,2 $\pm$ 3289,8 (14823) [70]
$CF_{14}$	(0) 0,8 $\pm$ 1,2 (4) [100]	(0) 25,3 $\pm$ 33,5 (111) [100]	(0) 993,4 $\pm$ 2015,2 (6988) [56,7]
$CF_{15}$	(0) 1,1 $\pm$ 1,4 (4) [100]	(0) 40,1 $\pm$ 53,7 (174) [100]	(0) 1435,5 $\pm$ 3721,6 (14823) [53,3]
$CF_{16}$	(0) 0,9 $\pm$ 1,3 (4) [100]	(0) 60,1 $\pm$ 57,2 (181) [100]	(0) 1007,2 $\pm$ 3280 (14827) [66,7]
$CF_{17}$	(0) 0,5 $\pm$ 0,8 (4) [100]	(2) 56,4 $\pm$ 69,1 (181) [100]	(11) 729 $\pm$ 1028 (3138) [53,3]
$CF_{18}$	(0) 1 $\pm$ 1,3 (4) [100]	(0) 35,1 $\pm$ 54,4 (179) [100]	(1) 2962,9 $\pm$ 8011,8 (33285) [66,7]
$CF_{19}$	(0) 1 $\pm$ 1,3 (4) [100]	(0) 36 $\pm$ 52,4 (181) [100]	(3) 5152,6 $\pm$ 8489,5 (23598) [50]
$CF_{20}$	(0) 0,7 $\pm$ 1,2 (4) [100]	(1) 35,4 $\pm$ 48,6 (160) [100]	(1) 753,6 $\pm$ 1206,1 (4576) [56,7]
$CF_{21}$	(0) 1,1 $\pm$ 1,6 (4) [100]	(0) 40 $\pm$ 62,8 (181) [100]	(0) 582,8 $\pm$ 989,6 (3638) [46,7]
$CF_{22}$	(0) 0,9 $\pm$ 1,2 (4) [100]	(0) 37,5 $\pm$ 59,6 (181) [100]	(0) 397,6 $\pm$ 912,5 (4061) [70]
$CF_{23}$	(0) 0,8 $\pm$ 1,4 (4) [100]	(0) 60,8 $\pm$ 76,1 (181) [100]	(0) 652,8 $\pm$ 969,2 (3208) [60]
$CF_{24}$	(0) 0,7 $\pm$ 1 (4) [100]	(0) 33,8 $\pm$ 52,9 (181) [100]	(18) 913,3 $\pm$ 1655,8 (6933) [66,7]

**Table 9** Number of Backtracks solving Sudoku problem with different strategies

Strategy	Sud 2	Sud 12
$S_1$	18	0
$S_2$	10439	1
$S_3$	4	2
$S_4$	18	0
$S_5$	2	3
$S_6$	6541	3
$S_7$	9	1
$S_8$	2	3
	(min) avg $\pm$ sd (max) [%]	(min) avg $\pm$ sd (max) [%]
Random	(3) 101,2 $\pm$ 71,9 (261) [100]	(0) 1,8 $\pm$ 1,2 (4) [100]
$CF_1$	(0) 5,5 $\pm$ 5,7 (23) [100]	(0) 1,4 $\pm$ 0,9 (3) [100]
$CF_2$	(0) 41,6 $\pm$ 184,8 (1019) [100]	(0) 0,8 $\pm$ 0,9 (2) [100]
$CF_3$	(1) 8,2 $\pm$ 16,1 (81) [100]	(0) 1,2 $\pm$ 0,9 (3) [100]
$CF_4$	(0) 12,9 $\pm$ 36,4 (203) [100]	(0) 1 $\pm$ 0,9 (2) [100]
$CF_5$	(0) 10,5 $\pm$ 20 (107) [100]	(0) 1,3 $\pm$ 0,8 (2) [100]
$CF_6$	(2) 6,1 $\pm$ 8,4 (45) [100]	(0) 1 $\pm$ 0,8 (2) [100]
$CF_7$	(0) 18,6 $\pm$ 34,3 (154) [100]	(0) 1,4 $\pm$ 0,9 (3) [100]
$CF_8$	(0) 8,1 $\pm$ 11,5 (58) [100]	(0) 1,3 $\pm$ 1 (3) [100]
$CF_9$	(1) 8,3 $\pm$ 13,6 (59) [100]	(0) 0,9 $\pm$ 0,9 (2) [100]
$CF_{10}$	(0) 7,4 $\pm$ 15 (82) [100]	(0) 1,4 $\pm$ 0,9 (3) [100]
$CF_{11}$	(0) 4,8 $\pm$ 5 (18) [100]	(0) 1,2 $\pm$ 1 (3) [100]
$CF_{12}$	(2) 11,5 $\pm$ 17,5 (74) [100]	(0) 1,2 $\pm$ 1 (3) [100]
$CF_{13}$	(0) 7,6 $\pm$ 16,5 (92) [100]	(0) 1,2 $\pm$ 0,9 (2) [100]
$CF_{14}$	(2) 43,8 $\pm$ 208,6 (1148) [100]	(0) 1,4 $\pm$ 1 (3) [100]
$CF_{15}$	(0) 10,1 $\pm$ 17,6 (86) [100]	(0) 1,4 $\pm$ 0,9 (3) [93,3]
$CF_{16}$	(0) 9,5 $\pm$ 16,2 (68) [100]	(0) 1,2 $\pm$ 0,9 (3) [100]
$CF_{17}$	(2) 23,9 $\pm$ 58,5 (322) [100]	(0) 1,1 $\pm$ 1,1 (3) [100]
$CF_{18}$	(0) 50,5 $\pm$ 209,9 (1158) [100]	(0) 1,4 $\pm$ 0,8 (3) [100]
$CF_{19}$	(1) 14,2 $\pm$ 19,4 (81) [100]	(0) 1,3 $\pm$ 0,9 (3) [100]
$CF_{20}$	(2) 7,8 $\pm$ 7,1 (25) [100]	(0) 1,2 $\pm$ 0,9 (2) [100]
$CF_{21}$	(1) 13,4 $\pm$ 23,5 (75) [100]	(0) 1,4 $\pm$ 1 (3) [100]
$CF_{22}$	(2) 120,8 $\pm$ 595,4 (3272) [100]	(0) 1,4 $\pm$ 0,9 (3) [100]
$CF_{23}$	(2) 12,5 $\pm$ 16,4 (67) [100]	(0) 1,3 $\pm$ 0,8 (2) [100]
$CF_{24}$	(1) 11,2 $\pm$ 18,9 (85) [100]	(0) 1,1 $\pm$ 1 (3) [100]

**Table 10** (a) Ranking with respect to the Number of Backtracks for N-Queens problem, (b) Ranking with respect of the Number of Backtracks for Sudoku problem, and (c) Ranking with respect of the Number of Backtracks for Sudoku problem.

Place	NQ n=8	NQ n=9	NQ n=10	NQ n=11	NQ n=12	NQ n=13	NQ n=14	NQ n=15	NQ n=16	NQ n=17	NQ n=18	NQ n=19	NQ n=20
1 <sub>st</sub>	CF10	S1	S5	S1	CF15	S1	S5	S7	S5	S3	S3	S5	S5
2 <sub>nd</sub>	CF5	S5	S7	S4	CF9	S4	S7	S7	S7	CF16	S7	S7	S7
3 <sub>rd</sub>	CF14	S3	S1	S5	S5	S5	CF18	CF9	CF18	CF16	CF13	S1	CF14
4 <sub>th</sub>	CF1	S4	S4	S8	S6	S8	CF23	CF5	CF16	CF13	CF16	S4	CF2
5 <sub>th</sub>	CF6	S5	S8	CF23	CF3	CF9	CF8	S1	CF4	CF4	CF11	S5	CF12
6 <sub>th</sub>	CF6	S6	S8	CF16	CF7	CF15	CF3	S4	CF1	CF14	CF1	S8	CF6
7 <sub>th</sub>	CF15	S7	CF20	CF12	CF11	CF12	CF16	S5	CF8	CF17	CF17	CF17	CF3
8 <sub>th</sub>	CF15	S8	CF5	CF4	CF10	CF11	CF4	S6	CF15	CF1	CF17	CF4	CF2
9 <sub>th</sub>	CF24	CF19	CF3	CF6	CF14	CF8	CF9	CF15	CF11	CF8	CF14	CF7	CF5
10 <sub>th</sub>	CF9	CF18	CF6	CF7	CF10	CF17	CF24	CF13	CF12	CF7	CF5	CF2	CF2
11 <sub>th</sub>	CF22	CF23	CF17	CF15	CF16	CF4	CF23	CF13	CF13	CF20	CF4	CF23	CF11
12 <sub>th</sub>	CF7	CF2	CF18	CF5	CF4	CF14	CF22	CF12	CF14	CF2	Random	CF19	CF4
13 <sub>th</sub>	CF13	CF7	CF5	CF13	CF6	CF11	CF18	CF16	CF5	CF9	CF2	CF18	CF10
14 <sub>th</sub>	CF1	CF22	CF7	S5	CF18	CF11	CF16	CF1	CF4	CF24	CF22	CF19	CF14
15 <sub>th</sub>	CF5	CF7	CF13	S7	CF22	CF10	CF1	CF20	CF2	CF12	CF12	CF1	CF16
16 <sub>th</sub>	CF12	CF17	CF10	CF22	CF12	CF7	CF20	CF2	CF10	CF17	CF8	CF9	S5
17 <sub>th</sub>	CF3	CF1	CF11	CF1	CF23	S3	CF23	CF8	CF24	CF15	CF15	CF3	S6
18 <sub>th</sub>	CF3	CF12	CF24	CF1	CF3	S7	CF14	CF11	CF19	CF21	CF10	CF9	Random
19 <sub>th</sub>	CF9	CF5	CF22	CF24	CF24	S7	CF14	CF11	CF19	CF21	CF10	CF9	CF9
20 <sub>th</sub>	CF4	CF16	CF23	CF9	CF1	CF3	CF9	CF18	CF6	CF5	CF3	CF20	CF8
21 <sub>th</sub>	CF16	CF10	CF16	CF10	S1	CF2	CF23	CF6	CF7	S1	CF22	CF10	CF13
22 <sub>th</sub>	CF18	CF15	CF21	CF17	S4	CF18	CF7	CF21	CF6	S4	CF20	CF8	CF17
23 <sub>th</sub>	CF17	CF8	CF8	CF3	S5	CF21	CF21	CF7	CF17	S5	CF6	CF22	CF9
24 <sub>th</sub>	CF20	CF4	CF19	CF19	S8	CF20	CF15	CF22	CF15	S8	CF23	CF15	CF15
25 <sub>th</sub>	Random	CF9	CF15	CF21	CF13	CF19	CF16	CF24	CF23	CF6	CF18	CF12	CF23
26 <sub>th</sub>	S1	CF1	CF12	CF8	CF21	CF24	CF17	CF14	CF21	CF3	CF21	CF15	CF18
27 <sub>th</sub>	S5	CF24	CF4	CF11	CF17	CF23	S5	CF5	S5	CF15	CF24	CF16	CF21
28 <sub>th</sub>	S4	CF21	CF1	CF14	CF20	CF6	S6	CF19	S6	CF11	S1	CF11	CF21
29 <sub>th</sub>	S7	CF13	CF14	CF20	S5	CF1	Random	CF23	Random	CF19	S4	CF21	CF24
30 <sub>th</sub>	S7	CF6	CF5	CF18	S7	CF5	S1	Random	S1	CF18	S8	CF6	S1
31 <sub>th</sub>	S8	CF14	Random	Random	Random	S4	CF17	S4	Random	S8	Random	S8	S1
32 <sub>th</sub>	S2	CF20	S5	S5	S5	S5	S5	S5	S5	S2	S2	S5	S5
33 <sub>th</sub>	S6	Random	S6	S6	S6	S6	S6	S6	S6	S6	S6	S6	S6

Place	MS n=3	MS n=4	MS n=5
1 <sub>st</sub>	S1	S1	S1
2 <sub>nd</sub>	S3	S4	CF8
3 <sub>rd</sub>	S4	S1	CF22
4 <sub>th</sub>	S6	CF14	CF21
5 <sub>th</sub>	CF12	S8	CF6
6 <sub>th</sub>	CF5	random	CF23
7 <sub>th</sub>	CF17	CF24	CF9
8 <sub>th</sub>	CF7	CF12	CF17
9 <sub>th</sub>	CF10	CF18	CF4
10 <sub>th</sub>	CF1	CF20	CF20
11 <sub>th</sub>	CF24	CF6	CF5
12 <sub>th</sub>	CF5	CF19	S1
13 <sub>th</sub>	random	CF22	CF24
14 <sub>th</sub>	CF20	CF3	CF2
15 <sub>th</sub>	CF6	CF1	CF14
16 <sub>th</sub>	CF23	CF21	CF16
17 <sub>th</sub>	CF7	CF15	CF10
18 <sub>th</sub>	CF8	S6	CF13
19 <sub>th</sub>	CF4	CF10	CF11
20 <sub>th</sub>	CF11	CF8	CF15
21 <sub>th</sub>	CF22	CF11	CF1
22 <sub>th</sub>	CF22	CF11	CF1
23 <sub>th</sub>	CF16	S8	CF7
24 <sub>th</sub>	CF18	CF13	CF12
25 <sub>th</sub>	CF19	CF1	CF18
26 <sub>th</sub>	S5	CF2	random
27 <sub>th</sub>	S7	CF6	CF19
28 <sub>th</sub>	S8	CF17	S4
29 <sub>th</sub>	CF14	CF16	S8
30 <sub>th</sub>	CF4	CF23	S6
31 <sub>th</sub>	CF15	CF7	S2
32 <sub>th</sub>	CF13	S7	S5
33 <sub>th</sub>	S2	S2	S7

Place	Sud	Sud
1 <sub>st</sub>	S5	S1
2 <sub>nd</sub>	S8	S4
3 <sub>rd</sub>	S5	CF2
4 <sub>th</sub>	CF11	CF9
5 <sub>th</sub>	CF1	CF4
6 <sub>th</sub>	CF6	CF6
7 <sub>th</sub>	CF10	S2
8 <sub>th</sub>	CF13	S7
9 <sub>th</sub>	CF20	CF17
10 <sub>th</sub>	CF8	CF24
11 <sub>th</sub>	CF3	CF20
12 <sub>th</sub>	CF9	CF13
13 <sub>th</sub>	S7	CF16
14 <sub>th</sub>	CF16	CF3
15 <sub>th</sub>	CF15	CF12
16 <sub>th</sub>	CF5	CF11
17 <sub>th</sub>	CF24	CF8
18 <sub>th</sub>	CF2	CF8
19 <sub>th</sub>	CF23	CF25
20 <sub>th</sub>	CF4	CF19
21 <sub>th</sub>	CF15	CF15
22 <sub>th</sub>	CF10	CF10
23 <sub>th</sub>	S1	CF1
24 <sub>th</sub>	S1	CF22
25 <sub>th</sub>	CF1	CF21
26 <sub>th</sub>	CF17	CF18
27 <sub>th</sub>	CF2	CF7
28 <sub>th</sub>	CF14	CF14
29 <sub>th</sub>	CF18	Random
30 <sub>th</sub>	Random	S3
31 <sub>th</sub>	CF22	S5
32 <sub>th</sub>	S6	S6
33 <sub>th</sub>	S2	S8

(c)

(b)

(a)

Our results agree with previous works showing good performance for strategies following the first-fail principle for variable selection (starting with those variables more difficult to assign values), strategies  $S_3$  and  $S_7$  in our work. Our decision-making proposal is a structured process and the research shows that its results are, on average, clearly better than the random selection of enumeration strategies. For all the problems one of our choice functions was at least in top-3 ranking, finding effective dynamic ways to improve the commutation of enumeration strategies during the search. This shows the ability of our approach to adapt itself to a different problem and that it could adapt itself and converge to an efficient strategy for the problem at hand. A global view of the search process can be obtained, by measuring its performance by means of some indicators, but such techniques do not take into account all the possible features. Many other features could be considered, but they interact in such unpredictable ways that it is often difficult to specify an adequate form of combining them. Our approach was able to combine dynamically some basic enumeration strategies using the information of some few indicators of the search.

The minimum value of backtracks reached in many experiments using our approach was zero (see **min** in Tables 7, 8, and 9). This demonstrates that it is possible to design “the best strategy” (or combination of enumeration strategies) for a specific problem.

## 4 Conclusions

In this paper we design a dynamic selection mechanism of enumeration strategies based on the information of the solving process. We did not focus on improving the resolution of a single problem, but we were interested in finding solutions good on average for a set of problems. The experimental results show that our proposal is able to consistently satisfy our requirements (of finding good solutions on average for a set of problems). We are at least in top-3 ranking finding good dynamic ways to solve many problems using a combination of enumeration strategies.

Among the main contributions of this work we can state the design and implementation of a solver that is able to measure the search process (using some basic indicators) in order to perform an on-the-fly replacement of enumeration strategies (using a portfolio of basic enumeration strategies). The solver is based on enumeration strategies of different natures (based on the size of variable domains, on the number of occurrences of the variables in the constraints) and some indicators on the resolution progress (backtracks, visited nodes, variables fixed, shallow backtracks, deep of the search tree, ...). These basic components work properly together using our framework. On the other hand, we have shown the development of a hybrid solver. In our approach the replacement of the enumeration strategies is performed depending on a quality rank (priority), which is computed by means of a choice function based hyperheuristic and its parameters are fine-tuned by a genetic algorithm. This tuning is a form of search too. We were focused on a general meta-level tuning of the adaptation mechanism. This could be seen as the resolution of an



optimization problem whose solution would contain the optimal configuration of the choice function.

The correct tuning of the choice function weights has a crucial effect on the ability of the solver to properly solve specific problems. This setting is required because every problem has different characteristics. Parameter (choice function weights) tuning was difficult to achieve because the parameters are problem dependent and the best values of parameters are not stable along the search. Therefore static weights lead to sub-optimal searches [12]. Moreover, parameters usually interact in a complex way, so a single parameter will have a different effect depending on the value of the others [12]. Then, combining the indicators in other form (non linear) could be an extension of this work. Additionally, this work may be extended implementing other choice functions and tuning their parameters with other meta-heuristics, benefiting from the addition of restarting, considering constraint propagation and tackling optimization problems.

## References

1. Barták, R., Rudová, H.: Limited assignments: A new cutoff strategy for incomplete depth-firstsearch. In: Haddad, H., Liebrock, L.M., Omicini, A., Wainwright, R.L. (eds.) SAC, pp. 388–392. ACM (2005)
2. Beck, J.C., Prosser, P., Wallace, R.J.: Toward understanding variable ordering heuristics for constraint satisfaction problems. In: Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS), pp. 11–16 (2003)
3. Bessière, C., Zanuttini, B., Fernandez, C.: Measuring search trees. In: Proceedings ECAI 2004 Workshop on Modelling and Solving Problems with Constraints, pp. 31–40. IOS Press (2004)
4. Borrett, J.E., Tsang, E.P.K., Walsh, N.R.: Adaptive constraint satisfaction: The quickest first principle. In: Wahlster, W. (ed.) ECAI, pp. 160–164. John Wiley and Sons, Chichester (1996)
5. Castro, C., Monfroy, E., Figueroa, C., Meneses, R.: An Approach for Dynamic Split Strategies in Constraint Solving. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICA 2005. LNCS (LNAI), vol. 3789, pp. 162–174. Springer, Heidelberg (2005)
6. Chenouard, R., Granvilliers, L., Sebastian, P.: Search heuristics for constraint-aided embodiment design. *AI EDAM* 23(2), 175–195 (2009)
7. Cowling, P.I., Kendall, G., Soubeiga, E.: A Hyperheuristic Approach to Scheduling a Sales Summit. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 176–190. Springer, Heidelberg (2001)
8. Crawford, B., Castro, C., Monfroy, E.: Using a choice function for guiding enumeration in constraint solving. In: 9th Mexican International Conference on Artificial Intelligence, MICA 2010, Pachuca, Mexico, November 8-13, Special Sessions, Revised Papers, pp. 37–42 (2010)
9. Hamadi, Y., Monfroy, E., Saubion, F.: What is autonomous search? Technical Report MSR-TR-2008-80, Microsoft Research (2008)
10. Bayardo Jr., R.J., Miranker, D.P.: An optimal backtrack algorithm for tree-structured constraint satisfaction problems. *Artif. Intell.* 71(1), 159–181 (1994)
11. Mackworth, A.K., Freuder, E.C.: The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artif. Intell.* 25(1), 65–74 (1985)

12. Maturana, J., Saubion, F.: From parameter control to search control: Parameter control abstraction in evolutionary algorithms. *Constraint Programming Letters* 4(1), 39–65 (2008)
13. Monfroy, E., Castro, C., Crawford, B.: Adaptive Enumeration Strategies and Metabacktracks for Constraint Solving. In: Yakhno, T., Neuhold, E.J. (eds.) *ADVIS 2006. LNCS*, vol. 4243, pp. 354–363. Springer, Heidelberg (2006)
14. Sadeh, N.M., Fox, M.S.: Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem. *Artif. Intell.* 86(1), 1–41 (1996)
15. Sturdy, P.: Learning Good Variable Orderings. In: Rossi, F. (ed.) *CP 2003. LNCS*, vol. 2833, p. 997. Springer, Heidelberg (2003)
16. Talbi, E.G.: *Metaheuristics: From Design to Implementation*. Wiley Publishing (2009)

# Maximum Parsimony Phylogenetic Inference Using Simulated Annealing

Jean-Michel Richer, Eduardo Rodriguez-Tello, and Karla E. Vazquez-Ortiz

**Abstract.** The Maximum Parsimony (MP) problem aims at reconstructing a phylogenetic tree from DNA sequences while minimizing the total number of genetic transformations. In this paper we propose a carefully devised simulated annealing implementation, called SAMPARS (Simulated Annealing for Maximum PARSimony), for finding near-optimal solutions for the MP problem. Different possibilities for its key components and input parameter values were carefully analyzed and tuned in order to find the combination of them offering the best quality solutions to the problem at a reasonable computational effort. Its performance is investigated through extensive experimentation over well known benchmark instances showing that our SAMPARS algorithm is able to improve some previous best-known solutions.

**Keywords:** Maximum Parsimony, Phylogenetic Trees, Simulated Annealing.

## 1 Introduction

One of the main problems in Comparative Biology consists in establishing ancestral relationships between a group of  $n$  species or homologous genes in populations of different species, designated as taxa. These ancestral relationships are usually represented by a binary rooted tree, which is called a phylogenetic tree or a phylogeny [19].

In the past phylogenetic trees were inferred by using morphologic characteristics like color, size, number of legs, etc. Nowadays, they are reconstructed using the

---

Jean-Michel Richer

LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers Cedex 01, France

e-mail: [jean-michel.richer@univ-angers.fr](mailto:jean-michel.richer@univ-angers.fr)

Eduardo Rodriguez-Tello and Karla E. Vazquez-Ortiz

CINVESTAV-Tamaulipas, Information Technology Laboratory,

Km. 5.5 Carretera Victoria-Soto La Marina, 87130 Victoria Tamps., Mexico

e-mail: [{kvazquez, ertello}@tamps.cinvestav.mx](mailto:{kvazquez, ertello}@tamps.cinvestav.mx)

information from biologic macromolecules like DNA (deoxyribonucleic acid), RNA (ribonucleic acid) and proteins. The problem of reconstructing molecular phylogenetic trees has become an important field of study in Bioinformatics and has many practical applications in population genetics, whole genome analysis, and the search for genetic predictors of disease [20, 35].

Given a set  $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$  composed by  $n$  sequences of length  $k$  over a predefined alphabet  $\mathcal{A}$  (operational taxa previously aligned). A *binary rooted phylogenetic tree*  $T = (V, E)$  is used to represent their ancestral relationships, it consists of a set of nodes  $V = \{v_1, \dots, v_r\}$  and a set of edges  $E \subseteq V \times V = \{\{u, v\} \mid u, v \in V\}$ . The set of nodes  $V$  ( $|V| = (2n - 1)$ ) is partitioned into two subsets:  $I$ , containing  $n - 1$  *internal nodes* (hypothetical ancestors) each one having 2 descendants; and  $L$ , composed of  $n$  *leaves*, i.e., nodes with no descendant.

The parsimony sequence  $P_w = \{z_1, \dots, z_k\}$  for each internal node  $I_w \in I$ , whose descendants are  $S_u = \{x_1, \dots, x_k\}$  and  $S_v = \{y_1, \dots, y_k\}$ , is calculated with the following expression:

$$\forall i, 1 \leq i \leq k, z_i = \begin{cases} x_i \cup y_i, & \text{if } x_i \cap y_i = \emptyset \\ x_i \cap y_i, & \text{otherwise} \end{cases}. \quad (1)$$

Then, the parsimony cost of the sequence  $P_w$  is defined as follows:

$$\phi(P_w) = \sum_{i=1}^k C_i \quad \text{where} \quad C_i = \begin{cases} 1, & \text{if } x_i \cap y_i = \emptyset \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

and the parsimony cost for the tree  $T$  is obtained as follows:

$$\phi(T) = \sum_{w \in I} \phi(P_w). \quad (3)$$

Thus, the Maximum Parsimony (MP) problem consists in finding a tree topology  $T^*$  for which  $\phi(T^*)$  is minimum, i.e.,

$$\phi(T^*) = \min\{\phi(T) : T \in \mathcal{T}\}, \quad (4)$$

where  $\mathcal{T}$  is the set composed by all the possible tree topologies (the search space of the problem).

There exist many different methods reported in the literature, to solve the problem of reconstructing phylogenetic trees. These can be classified in three main different approaches. *Distance methods* [13, 32], *Probabilistic methods* [12, 33] and *Cladistic methods* [11, 7]. In this paper we focus our attention in a cladistic method based on the Maximum Parsimony (MP) criterion, which is considered in the literature as one of the most suitable evaluation criterion for phylogenies [24, 34].

It has been demonstrated that the MP problem is NP-complete [17], since it is equivalent to the combinatorial optimization problem known as the Steiner tree problem on hypercubes, which is proven to be NP-complete [14].

The MP problem has been exactly solved for very small instances ( $n \leq 10$ ) using a branch & bound algorithm (B&B) originally proposed by Hendy and Penny [18]. However, this algorithm becomes impractical when the number of studied species  $n$  increases, since the size of the search space suffers a combinatorial explosion. Therefore, there is a need for heuristic methods to address the MP problem in reasonable time.

Andreatta and Ribeiro [4] compared three greedy algorithms of different complexity: *IstRotuGbr*, *Gstep\_wR* and *Grstep*. They concluded from their experiments that, *Gstep\_wR* was more efficient than *IstRotuGbr*, but expending more computational time. *Grstep* achieved good results only when it was combined with a local search method. Even when these methods attained good quality solutions, they were still far away from the optimal solutions.

In 2003, Barker proposed a software, called LVB, which implemented a multi-start simulated annealing algorithm for solving the MP problem [5]. Later, an updated version of LVB was released in 2010 [6]. This new version adds a hill-climbing phase at the end of each simulated annealing search and a new stop condition.

Ribeiro and Viana [26] in 2005 applied a greedy randomized adaptive search procedure (GRASP) for solving the MP problem and showed that this algorithm had the best performance with respect to the state-of-the-art algorithms. Different evolutionary algorithms were also reported for the MP problem. Among them we found GA+PR+LS, a genetic algorithm hybridized with local search which employs path-relinking to implement a progressive crossover operator [27]. More recently Richer, Goëffon and Hao [28] introduced a memetic algorithm called Hydra which yields the best-known solutions for a set of 20 benchmark instances proposed in [25].

This paper aims at developing a new simulated annealing (SA) algorithm implementation (hereafter called SAMPARS) for finding near-optimal solutions for the MP problem under the Fitch's criterion. To achieve this, different possibilities for its key components were carefully designed and evaluated. The SAMPARS input parameter values yielding the best quality solutions to the problem at a reasonable computational effort were determined by employing a tuning methodology based on Combinatorial Interaction Testing. The performance of the new proposed implementation is investigated through extensive experimentation over 20 well known benchmark instances and compared with other existing state-of-the-art algorithms, showing that our algorithm is able to improve some previous best-known solutions.

The rest of this paper is organized as follows. In Section 2 the components of our SAMPARS implementation are discussed in detail. Then, three computational experiments are presented in Sect. 3 devoted to determine the best parameter settings for SAMPARS and to compare its performance with respect to LVB, an existing SA implementation [5, 6] and two other representative state-of-the-art algorithms: GA+PR+LS [27] and Hydra [15]. Finally, the last section summarizes the main contributions of this work and presents some possible directions for future research.

## 2 An Improved Implementation of a Simulated Annealing Algorithm

Simulated Annealing (SA) is a general-purpose stochastic optimization technique that has proved to be an effective tool for approximating globally optimal solutions to many NP-hard optimization problems. However, it is well known that developing an effective SA algorithm requires a careful implementation of some essential components and an appropriate tuning of the parameters used [21, 22].

In this section we present an improved implementation of a SA algorithm (Algorithm 1), that we called SAMPARS, for finding tree topologies (phylogenies) with near-optimal parsimony costs under the Fitch's criterion. The main difference of our implementation, with respect to LVB [5, 6], occurs in the neighborhood function (line 7) which has been tailored to fit the specificity of the MP problem. SAMPARS employs a composed neighborhood function combining standard neighborhood relations for trees with a stochastic descent algorithm on the current solution (see Sect. 2.4), while LVB randomly selects a neighbor  $T' \in \mathcal{T}$  of the current solution  $T$ .

---

### Algorithm 1 SAMPARS algorithm

---

```

input:  $\mathcal{N}$ ,  $\phi$ ,  $t_i$ ,  $CL$ ,  $\alpha$ ,  $\beta$ 
 $T \leftarrow \text{GenerateInitialSolution}()$ 
 $T^* \leftarrow T$ 
 $j \leftarrow 0$ 
 $t_j \leftarrow t_i$ 
repeat
   $c \leftarrow 0$ 
  while  $c < CL$  do
     $c \leftarrow c + 1$ 
     $T' \leftarrow \text{GenerateNeighbor}(T, c, \mathcal{N})$ 
     $\Delta\phi \leftarrow \phi(T') - \phi(T)$ 
    Generate a random  $u \in [0, 1]$ 
    if  $(\Delta\phi < 0)$  or  $(e^{-\Delta\phi/t} > u)$  then
       $T \leftarrow T'$ 
      if  $\phi(T') < \phi(T^*)$  then  $T^* \leftarrow T'$ 
     $j \leftarrow j + 1$ 
     $t_j \leftarrow \text{UpdateCurrentTemperature}(t_{j-1})$ 
until  $< \text{stop condition} >$ 
return  $T^*$ 

```

---

Next all the implementation details of the proposed SAMPARS algorithm are presented. For some of these components different possibilities were analyzed

(see Sect. 3.2) in order to find the combination of them which offers the best quality solutions at a reasonable computational effort.

## 2.1 Internal Representation and Search Space

Let  $T$  be a potential solution in the search space  $\mathcal{T}$ , that is a phylogenetic tree representing the ancestral relationships of a group of  $n$  operational taxa, each one of length  $k$  over a predefined alphabet  $\mathcal{A}$ . Then  $T$  is represented as a binary rooted tree composed of  $n - 1$  internal nodes and  $n$  leaves. The size of the search space  $|\mathcal{T}|$ , i.e., the number of rooted tree topologies is given by the following expression [40]:

$$|\mathcal{T}| = (2n - 3)! / 2^{n-2} (n - 2)! \quad (5)$$

## 2.2 Evaluation Function

The evaluation function is one of the key elements for the successful implementation of metaheuristic algorithms because it is in charge of guiding the search process toward good solutions in a combinatorial search space.

Previously reported metaheuristic methods for solving the MP problem have commonly evaluated the quality of a potential solution,  $\phi(T)$ , using the parsimony cost depicted in (3) [4, 5, 26, 27, 28]. In our SAMPARS implementation this evaluation function was also used.

## 2.3 Initial Solution

The initial solution is the starting phylogenetic tree used for the algorithm to begin the search of better configurations in the search space  $\mathcal{T}$ .

In the existing SA implementation for the MP problem [5, 6] the initial solution is randomly generated. In contrast, SAMPARS creates the starting solution using a greedy procedure that guarantees a better quality initial solution. The proposed procedure can be described as follows.

First, it generates a random permutation of the studied taxa, which is used to indicate the order in which the leaves (taxa) will be processed. Then, the root node of the tree is created and the first and second taxa in the permutation are binded to it. The rest of the taxa in the permutation are appended to the tree one by one. Each time a new taxon is added to the partial tree, the algorithm analyzes all the possible insertion positions in order to select the one that minimizes the increase in the tree's parsimony cost. This process is iterated until all the remaining taxa in the permutation are processed.

## 2.4 Neighborhood Functions

The most common practice in the reported metaheuristics for the MP problem [4, 26, 27] is employing one of the following three neighborhood functions. The first one, called Nearest Neighbor Interchange (NNI), was proposed by Waterman and Smith [39]. It exchanges two subtrees separated by an internal node. Given that each tree has  $n - 3$  internal nodes and two possible moves by branch, then there exist  $2n - 6$  NNI neighboring solutions [29]. The second one, is known as Subtree Pruning and Regrafting (SPR) [36]. It cuts a branch of the tree and reinserts the resulting subtree elsewhere generating a new internal node. For each tree there exist  $2(n - 3)(2n - 7)$  possible SPR neighboring solutions [3]. Finally, the Tree Bisection and Reconnection (TBR) [36] consists in dividing the tree into two subtrees that will be reconnected from one of their branches. From a given tree, the TBR neighborhood induces at most  $(2n - 3)(n - 3)^2$  neighboring trees [3].

LVB, the existing SA implementation for the MP problem [5, 6] alternates the use of the NNI and SPR neighborhood functions at each iteration of the search process. In the case of our SAMPARS algorithm both the SPR and the TBR neighborhood relations are implemented. However, from our preliminary experiments it has been observed that the separated use of these neighborhood functions is not sufficient to reach the best-known solutions, because both of them are highly disruptive. In order to achieve a better performance for SAMPARS, we have decided to use a third complementary neighborhood structure. It is based on a stochastic descent algorithm with a best-improve scheme which is occasionally applied to the neighboring solution  $T'$  prior to returning it. Our compound neighborhood function is inspired by the ideas reported in [23], where the advantage of using this approach is well documented.

---

### Algorithm 2 GenerateNeighbor

---

**input:**  $T, c, \mathcal{N}$   
 randomly select  $T' \in \mathcal{N}(T)$  // A SPR or TBR neighboring solution  
**if**  $c$  is a multiple of 15 **then**  
      $T' \leftarrow \text{Descent}(T')$   
**return**  $T'$

---

## 2.5 Cooling Schedule

A cooling schedule is defined by the following parameters: an initial temperature, a final temperature or a stopping criterion, the maximum number of neighboring solutions that can be generated at each temperature (Markov chain length), and a rule for decrementing the temperature. The cooling schedule governs the convergence of



the SA algorithm. At the beginning of the search, when the temperature is large, the probability of accepting solutions of worse quality than the current solution (uphill moves) is high. It allows the algorithm to escape from local minima. The probability to accept such moves is gradually decreased as the temperature goes to zero.

The literature offers a number of different cooling schedules, see for instance [1, 37, 2, 30]. They can be divided into two main categories: static and dynamic. In a static cooling schedule, the parameters are fixed and cannot be changed during the execution of the algorithm. With a dynamic cooling schedule the parameters are adaptively changed during the execution.

In our SAMPARS implementation we preferred a geometrical cooling scheme (static) mainly for its simplicity. It starts at an initial temperature  $t_i$  that can either be defined by the user or automatically computed using the following formula:  $(k+n)^{(1.0/3.3)}$  which generates values under 6.0 for most of the tested benchmark instances. Then, this temperature is decremented at each round by a factor  $\alpha = 0.99$  using the relation  $t_j = \alpha t_{j-1}$ . A reheat mechanism has also been implemented. If the best-so-far solution is not improved during  $maxNITD = 50$  consecutive temperature decrements, the current temperature  $t_j$  is increased by a factor  $\beta = 1.4$  using the function  $t_j = \beta t_{j-1}$ . In our implementation this reheat mechanism can be applied at most  $maxReheat = 3$  times, since it represents a good trade-off between efficiency and quality of solution found.

For each temperature  $t_j$ , the maximum number of visited neighboring solutions is  $CL$ . It depends directly on the parameters  $n$  and  $k$  of the studied instance, since we have observed that more moves are required for bigger trees [38]. Next, we present the three different values that  $CL$  can take.

- small:  $CL = 15(n+k)$
- medium:  $CL = 23(n+k)$
- large:  $CL = 40(n+k)$

The parameter values presented in this section were chosen based on the results obtained in a preliminary experimentation. For the reason of space limitation we did not present here these experiments.

## 2.6 Stop Condition

The SAMPARS algorithm stops if it ceases to make progress. In our implementation a lack of progress exists if after  $\omega = 40$  (*frozen factor*) consecutive temperature decrements the best-so-far solution is not improved.

We will see later that thanks to the main features presented in this section, our SAMPARS algorithm reaches good quality results, which are sometimes better than the best-known solutions reported in the literature [15, 27].

### 3 Computational Experiments

In this section three main experiments were accomplished to evaluate the performance of the proposed SAMPARS algorithm and some of its components are presented. The objective of the first experiment is to determine both a component combination, and a set of parameter values which permit SAMPARS to attain the best trade-off between solution quality and computational effort. The purpose of the second experiment is to carry out a performance comparison of SAMPARS with respect to an existing SA algorithm called LVB [5, 6]. The third experiment is devoted to assess the performance of SAMPARS with respect to two representative state-of-the-art procedures: Hydra [15] and GA+PR+LS [27].

For these experiments SAMPARS was coded in C++ and compiled with `g++` using the optimization flag `-O3`. It was run sequentially into a CPU Xeon X5650 at 2.66 GHz, 2 GB of RAM with Linux operating system. Due to the non-deterministic nature of the studied algorithms, 30 independent runs were executed for each of the selected benchmark instances in each experiment presented in this section.

#### 3.1 Benchmark Instances and Performance Assessment

The test-suite that we have used in our experiments is the same proposed by Ribeiro and Vianna [26] and later employed in other works [15, 27]. It consists of 20 randomly generated instances with a number of sequences ( $n$ ) ranging from 45 to 75 whose length ( $k$ ) varies from 61 to 159.

For all the experiments, 30 independent executions were performed. The criteria used for evaluating the performance of the algorithms are the same as those used in the literature: the best parsimony cost found for each instance (smaller values are better) and the expended CPU time in seconds.

#### 3.2 Components and Parameters Tuning

Optimizing parameter settings is an important task in the context of algorithm design. Different procedures have been proposed in the literature to find the most suitable combination of parameter values [10, 16]. In this paper we employ a tuning methodology based on Combinatorial Interaction Testing (CIT) [8]. We have decided to use CIT, because it allows to significantly reduce the number of tests (experiments) needed to determine the best parameter settings of an algorithm. Instead of exhaustive testing all the parameter value combinations of the algorithm, it only analyzes the interactions of  $t$  (or fewer) input parameters by creating interaction test-suites that include at least once all the  $t$ -way combinations between these parameters and their values.

**Table 1** Input parameters of the SAMPARS algorithm and their selected values

$IS$	$\alpha$	$t_i$	$\mathcal{N}$	$CL$
Greedy	0.99	6.0	SPR	$15(n+k)$
Random	0.85	$(k+n)^{(1,0/3,3)}$	TBR	$23(n+k)$
-	-	-	-	$40(n+k)$

Covering arrays (CAs) are combinatorial designs which are extensively used to represent those interaction test-suites. A covering array,  $CA(N; t, k, v)$ , of size  $N$ , strength  $t$ , degree  $k$ , and order  $v$  is an  $N \times k$  array on  $v$  symbols such that every  $N \times t$  sub-array includes, at least once, all the ordered subsets from  $v$  symbols of size  $t$  ( $t$ -tuples) [9]. The minimum  $N$  for which a  $CA(N; t, k, v)$  exists is the *covering array number* and it is defined according to the following expression:  $CAN(t, k, v) = \min\{N : \exists CA(N; t, k, v)\}$ .

CAs are used to represent an interaction test-suite as follows. In an algorithm we have  $k$  input parameters. Each of these has  $v$  values or levels. An interaction test-suite is an  $N \times k$  array where each row is a test case. Each column represents an input parameter and a value in the column is the particular configuration. This test-suite allows to cover all the  $t$ -way combinations of input parameter values at least once. Thus, the costs of tuning the algorithm can be substantially reduced by minimizing the number of test cases  $N$  in the covering array.

In practice, algorithms' input parameters do not have exactly the same number of values (levels). To overcome this limitation of CAs, mixed level covering arrays (MCAs) are used. A  $MCA(N; t, k, (v_1, v_2, \dots, v_k))$  is an  $N \times k$  array on  $v$  symbols ( $v = \sum_{i=1}^k v_i$ ), where each column  $i$  ( $1 \leq i \leq k$ ) of this array contains only elements from a set  $S_i$ , with  $|S_i| = v_i$ . This array has the property that the rows of each  $N \times t$  sub-array cover all  $t$ -tuples of values from the  $t$  columns at least once. Next, we present the details of the tuning process, based on CIT, for the particular case of our SAMPARS algorithm.

First, we have identified  $k = 5$  input parameters used for SAMPARS: initial solution procedure  $IS$ , cooling factor  $\alpha$ , initial temperature  $t_i$ , neighborhood function  $\mathcal{N}$  and maximum number of visited neighboring solutions  $CL$ . Based on some preliminary experiments, certain reasonable values were selected for each one of those input parameters (shown in Table 1).

The smallest possible mixed level covering array  $MCA(24; 4, 5, (2, 2, 2, 2, 3))$ , shown (transposed) in Table 2, was obtained by using the Memetic Algorithm reported in [31]. This covering array can be easily mapped into an interaction test-suite by replacing each symbol from each column to its corresponding parameter value. For instance, we can map 0 in the first column (the first line in Table 2) to Greedy and 1 to Random. The resulting interaction test-suite contains, thus, 24 test cases (parameter settings) which include at least once all the 4-way combinations between SAMPARS's input parameters and their values [4].

Each one of those 24 test cases was used to executed 30 times the SAMPARS algorithm over the 20 instances of the test-suite described in Sect. 3.1. The data

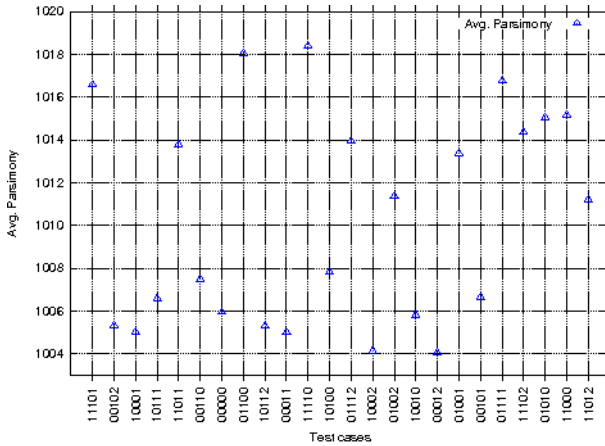
<sup>1</sup> In contrast, with an exhaustive testing which contains  $3(2^4) = 48$  test cases.

**Table 2** Mixed level covering array MCA(24;4,5,(2,2,2,2,3)) representing an interaction test-suite for tuning SAMPARS (transposed)

1	0	1	1	1	0	0	0	1	0	1	1	0	1	0	1	0	0	0	0	1	0	1	1
1	0	0	0	1	0	0	1	0	0	1	0	1	0	1	0	0	1	0	1	1	1	1	1
1	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	1	0	1	0	1
1	2	1	1	1	0	0	0	2	1	0	0	2	2	2	0	2	1	1	1	2	0	0	2

**Table 3** Results from the 5 best parameter test cases in the tuning experiment

Num.	Test case	Avg. parsimony	Avg. CPU time
<b>17</b>	<b>00012</b>	<b>1004.06</b>	<b>3512.51</b>
14	10002	1004.14	3511.35
10	00011	1005.00	2058.93
3	10001	1005.02	2047.52
9	10112	1005.29	3136.40



**Fig. 1** Average results obtained in the tuning experiments using 24 test cases over 20 standard benchmark instances

generated by these 14400 executions is summarized in Fig. 1 which depicts the average cost reached over the selected instances by each test case.

From this graphic we have selected the 5 test cases which yield the best results. Their average parsimony cost and the average CPU time in seconds are presented in Table 3. This table allowed us to observe that the parameter setting giving the best trade-off between solution quality and computational effort corresponds to the test case number 17 (shown in bold). The best average parsimony cost with an acceptable speed is thus reached with the following input parameter values: initial solution procedure  $IS = Greedy$ , cooling factor  $\alpha = 0.99$ , initial temperature  $t_i = 6.0$ , neighborhood function  $\mathcal{N} = TBR$  and maximum number of visited neighboring solutions  $CL = 40(n + k)$ . These values are thus used in the experimentation reported next.

### 3.3 Comparing SAMPARS with an Existing SA Implementation

For this experiment a subset of six representative benchmark instances, taken from the test-suite described in Sect. 3.1, was selected (comparable results were obtained with all the other tested instances). Then, the last version of LVB was obtained, compiled and executed in our computational platform using the input parameters suggested by their authors [6]. SAMPARS were also executed over the same six instances.

Table 4 displays the detailed computational results produced by this experiment. The first three columns in the table indicate the name of the instance as well as its number of taxa ( $n$ ) and length ( $k$ ). For each compared algorithm the best ( $B$ ), average ( $Avg.$ ), and standard deviation ( $Dev.$ ) of the parsimony cost attained in 30 independent executions and its average CPU time in seconds are listed in columns 4 to 11. A statistical significance analysis was performed for this experiment. First, *D’Agostino-Pearson’s omnibus  $K^2$*  test was used to evaluate the normality of data distributions. For normally distributed data, either *ANOVA* or the *Welch’s  $t$*  parametric tests were used depending on whether the variances across the samples were homogeneous (*homoskedasticity*) or not. This was investigated using the *Bartlett’s* test. For non-normal data, the nonparametric *Kruskal-Wallis* test was adopted. A significance level of 0.05 has been considered. The resulting *P-value* is presented in Column 12. Last column shows a “+” symbol if there exists a statistically significant increase in performance achieved by SAMPARS with regard to LVB, the existing SA Implementation.

**Table 4** Comparison between SAMPARS and LVB (an existing SA implementation [5, 6]) over a subset of six selected instances

Instance	$n$	$k$	LVB				SAMPARS				$P$ -value	SS
			$B$	$Avg.$	$Dev.$	$T$	$B$	$Avg.$	$Dev.$	$T$		
tst01	45	61	549	553.87	2.47	85.57	545	545.83	0.87	295.80	1.80E-11	+
tst02	47	151	1367	1375.33	4.77	23.63	1354	1356.13	1.33	479.50	5.08E-21	+
tst03	49	111	840	850.83	4.86	68.02	833	834.00	1.05	577.12	1.38E-18	+
tst08	57	119	867	879.80	5.01	922.61	852	854.53	2.37	665.50	2.12E-11	+
tst09	59	93	1153	1160.77	4.60	58.53	1143	1145.50	1.11	719.24	3.48E-18	+
tst10	60	71	727	738.00	5.59	570.62	720	721.27	0.78	500.28	1.89E-16	+
Avg.			917.17	926.43	4.55	288.16	907.83	909.54	1.25	539.57		

From Table 4 we can observe that SAMPARS is the most time-consuming algorithm, since it uses an average of 539.57 seconds for solving these six instances. On the contrary, LVB employs only 288.16 seconds. However, we can also remark that SAMPARS can take advantage of its longer executions. Indeed it is able to consistently improve the best results found by LVB, obtaining in certain instances, like *tst08*, an important decrease in the parsimony cost (up to  $-15 = 852 - 867$ ). Furthermore, the solution cost found by SAMPARS presents a relatively small standard deviation (see Column  $Dev.$ ). It is an indicator of the algorithm’s precision and

robustness since it shows that in average the performance of SAMPARS does not present important fluctuations.

The statistical analysis presented in the last two columns of Table 4 confirms that there exists a statistically significant increase in performance achieved by SAMPARS with regard to LVB on the six studied instances. Thus, we can conclude that SAMPARS is more effective than the existing SA algorithm reported in [5, 6]. Below, we will present more computational results obtained from a performance comparison carried out between SAMPARS and some state-of-the-art procedures.

### 3.4 Comparing SAMPARS with the State-of-the-Art Procedures

In this experiment a performance comparison of the best solutions achieved by SAMPARS with respect to those produced by GA+PR+LS [27] and Hydra [15] was carried out over the test-suite described in Sect. 3.1. The results from this experiment are depicted in Table 5. Columns 1 to 3 indicate the instance and its size in terms of taxa ( $n$ ) and length ( $k$ ). The best solutions found by GA+PR+LS and Hydra, in terms of parsimony cost  $\Phi$  are listed in the next two columns. Columns 6 to 9 present the best ( $B$ ), average ( $Avg.$ ), and standard deviation ( $Dev.$ ) of the parsimony cost attained by SAMPARS in 30 independent executions, as well as its average CPU time in seconds. Finally, the difference ( $\delta$ ) between the best result produced by our SAMPARS algorithm and the best-known solution produced by either GA+PR+LS or Hydra is shown in the last column.

**Table 5** Performance comparison among SAMPARS, GA+PR+LS [27] and Hydra [15] over 20 standard benchmark instances

Instance	$n$	$k$	GA+PR+LS	Hydra	SAMPARS				$\delta$
					$B$	$Avg.$	$Dev.$	$T$	
tst01	45	61	547	545	545	545.13	0.43	1407.57	0
tst02	47	151	1361	1354	1354	1355.30	0.97	1938.23	0
tst03	49	111	837	833	833	833.43	0.56	2506.30	0
tst04	50	97	590	588	<b>587</b>	588.23	0.80	1341.17	-1
tst05	52	75	792	789	789	789.00	0.00	2007.90	0
tst06	54	65	603	596	596	596.57	0.56	1164.27	0
tst07	56	143	1274	1269	1269	1270.83	1.63	4063.80	0
tst08	57	119	862	852	852	853.33	1.27	2884.73	0
tst09	59	93	1150	1144	<b>1141</b>	1144.73	1.09	3237.53	-3
tst10	60	71	722	721	<b>720</b>	720.80	0.70	2288.00	-1
tst11	62	63	547	542	<b>541</b>	542.21	0.72	3807.79	-1
tst12	64	147	1225	1211	<b>1208</b>	1215.27	2.76	3668.40	-3
tst13	65	113	1524	1515	1515	1517.77	1.91	2514.20	0
tst14	67	99	1171	1160	1160	1163.03	1.82	2847.13	0
tst15	69	77	758	752	752	753.90	1.11	4808.63	0
tst16	70	69	537	529	529	531.00	1.23	3268.20	0
tst17	71	159	2469	2453	<b>2450</b>	2456.00	2.63	8020.23	-3
tst18	73	117	1531	1522	<b>1521</b>	1525.67	3.96	4451.37	-1
tst19	74	95	1024	1013	<b>1012</b>	1016.23	2.14	6875.30	-1
tst20	75	79	671	661	<b>659</b>	662.82	1.44	7149.43	-2
Avg.			1009.75	1002.45	1001.65	1004.06	1.39	3512.51	

The analysis of the data presented in Table 5 lead us to the following observations. First, we clearly remark that the procedure GA+PR+LS [27] consistently returns poorer quality solutions than Hydra and SAMPARS. Second, the best solution quality attained by the proposed SAMPARS algorithm is very competitive with respect to that produced by the existing state-of-the-art procedure called Hydra [15], since in average SAMPARS provides solutions whose parsimony costs are smaller (compare Columns 5 and 6). In fact, it is able to improve on 9 previous best-known solutions produced by Hydra and to equal these results for the other 11 benchmark instances.

Thus, as this experiment confirms, our SAMPARS algorithm is an effective alternative for solving the MP problem, compared with the two more representative state-of-the-art algorithms: GA+PR+LS [27] and Hydra [15].

## 4 Conclusions

In this paper we have presented an improved simulated annealing algorithm (SAMPARS) for finding near-optimal solutions for the MP problem under the Fitch's criterion. SAMPARS's components and parameter values were carefully determined, through the use of a tuning methodology based on Combinatorial Interaction Testing [8], to yield the best solution quality in a reasonable computational time.

Extensive experimentation was conducted to investigate the performance of SAMPARS over a set of 20 well known benchmark instances. In these experiments our algorithm was carefully compared with an existing simulated annealing implementation (LVB) [5, 6], and other two state-of-the-art algorithms. The results show that there exists a statistically significant increase in performance achieved by SAMPARS with respect to LVB. SAMPARS is in fact able to consistently improve the best results produced by LVB, obtaining in certain instances important reductions in the parsimony cost. Regarding GA+PR+LS [27], we have observed that in average this algorithm returns worse quality solutions than SAMPARS. Compared with the state-of-the-art algorithm called Hydra [15], our SAMPARS algorithm was able to improve on 9 previous best-known solutions and to equal these results on the other 11 selected benchmark instances. Furthermore, it was observed that the solution cost found by SAMPARS presents a relatively small standard deviation, which indicates the precision and robustness of the proposed approach. These experimental results confirm the practical advantages of using our algorithm for solving the MP problem.

Finding near-optimal solutions for the MP problem is a very challenging problem. However, the introduction of SAMPARS opens up an exciting range of possibilities for future research. One fruitful possibility is to analyze the use of different cooling schedules, stop conditions and mechanism for adapting the maximum number of visited neighboring solutions at each temperature depending on the behavior of the search process.

**Acknowledgements.** This research work was partially funded by the following projects: 51623 Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas; CONACyT 99276, Algoritmos para la Canonizacin de Covering Arrays.

## References

1. Aarts, E.H.L., Van Laarhoven, P.J.M.: Statistical cooling: A general approach to combinatorial optimization problems. *Philips Journal of Research* 40, 193–226 (1985)
2. Abramson, D., Krishnamoorthy, M., Dang, H.: Simulated annealing cooling schedules for the school timetabling problem. *Asia-Pacific Journal of Operational Research* 16(1), 1–22 (1999)
3. Allen, B.J., Steel, M.: Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics* 5(1), 1–15 (2001)
4. Andreatta, A., Ribeiro, C.C.: Heuristics for the phylogeny problem. *Journal of Heuristics* 8(4), 429–447 (2002)
5. Barker, D.: LVB: parsimony and simulated annealing in the search for phylogenetic trees. *Bioinformatics* 20(2), 274–275 (2003)
6. Barker, D.: LVB homepage (2012), <http://biology.st-andrews.ac.uk/cegg/lvb.aspx>
7. Cavalli-Sforza, L.L., Edwards, A.W.F.: Phylogenetic analysis. models and estimation procedures. *The American Journal of Human Genetics* 19(3,pt 1), 233–257 (1967)
8. Cohen, D.M., Dalal, S.R., Parelius, J., Patton, G.C.: The combinatorial design approach to automatic test generation. *IEEE Software* 13(5), 83–88 (1996)
9. Colbourn, C.J.: Combinatorial aspects of covering arrays. *Le Matematiche* 58, 121–167 (2004)
10. de Landgraaf, W.A., Eiben, A.E., Nannen, V.: Parameter calibration using meta-algorithms. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 71–78. IEEE Press (2007)
11. Edwards, A.W.F., Cavalli-Sforza, L.L.: The reconstruction of evolution. *Heredity* 18, 553 (1963)
12. Felsenstein, J.: Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* 17(6), 368–376 (1981)
13. Fitch, W.M., Margoliash, E.: A method for estimating the number of invariant amino acid coding positions in a gene using cytochrome c as a model case. *Biochemical Genetics* 1(1), 65–71 (1967)
14. Garey, M.R., Johnson, D.S.: The rectilinear Steiner tree problem is NP-Complete. *SIAM Journal on Applied Mathematics* 32(4), 826–834 (1977)
15. Goëffon, A.: Nouvelles heuristiques de voisinage et mémétiques pour le problème maximum de parcimonie. Ph.D. thesis, LERIA, Université d'Angers (2006)
16. Gunawan, A., Lau, H.C., Lindawati: Fine-tuning algorithm parameters using the design of experiments. *LNCS*, vol. 6683, pp. 131–145 (2011)
17. Gusfield, D.: Algorithms on strings, trees, and sequences: Computer science and computational biology, 1st edn. Cambridge University Press (1997)
18. Hendy, M.D., Penny, D.: Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences* 59(2), 277–290 (1982)
19. Hennig, W.: Phylogenetic systematics. *Phylogeny*. University of Illinois Press, Urbana (1966)
20. Hillis, D.M., Moritz, C., Mable, B.K.: *Molecular systematics*, 2nd edn. Sinauer Associates Inc., Sunderland (1996)
21. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *Operations Research* 37(6), 865–892 (1989)
22. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. *Operations Research* 39(3), 378–406 (1991)



23. Lü, Z., Hao, J.K., Glover, F.: Neighborhood analysis: A case study on curriculum-based course timetabling. *Journal of Heuristics* 17(2) (2011)
24. Penny, D., Foulds, L.R., Hendy, M.D.: Testing the theory of evolution by comparing phylogenetic trees constructed from five different protein sequences. *Nature* 297, 197–200 (1982)
25. Ribeiro, C.C., Vianna, D.S.: A genetic algorithm for the phylogeny problem using an optimized crossover strategy based on path-relinking. In: *Proceedings of the II Workshop Brasileiro de Bioinformática, Macaé, Brazil*, pp. 97–102 (2003)
26. Ribeiro, C.C., Vianna, D.S.: A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research* 12(3), 325–338 (2005)
27. Ribeiro, C.C., Vianna, D.S.: A hybrid genetic algorithm for the phylogeny problem using path-relinking as a progressive crossover strategy. *International Transactions in Operational Research* 16(5), 641–657 (2009)
28. Richer, J.-M., Goëffon, A., Hao, J.-K.: A Memetic Algorithm for Phylogenetic Reconstruction with Maximum Parsimony. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) *EvoBIO 2009. LNCS*, vol. 5483, pp. 164–175. Springer, Heidelberg (2009)
29. Robinson, D.F.: Comparison of labeled trees with valency three. *Journal of Combinatorial Theory, Series B* 11(2), 105–119 (1971)
30. Rodriguez-Tello, E., Hao, J.K., Torres-Jimenez, J.: An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *Computers & Operations Research* 35(10), 3331–3346 (2008)
31. Rodriguez-Tello, E., Torres-Jimenez, J.: Memetic Algorithms for Constructing Binary Covering Arrays of Strength Three. In: Collet, P., Monmarché, N., Legrand, P., Schoenauer, M., Lutton, E. (eds.) *EA 2009. LNCS*, vol. 5975, pp. 86–97. Springer, Heidelberg (2010)
32. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* 4(4), 406–425 (1987)
33. Skourikhine, A.: Phylogenetic tree reconstruction using self-adaptive genetic algorithm. In: *Proceedings of the IEEE International Symposium on Bio-Informatics and Biomedical Engineering*, Arlington, VA, USA, pp. 129–134 (2000)
34. Sober, E.: *The nature of selection: Evolutionary theory in philosophical focus*. University Of Chicago Press (1993)
35. Sridhar, S., Lam, F., Belloch, G.E., Ravi, R., Schwartz, R.: Direct maximum parsimony phylogeny reconstruction from genotype data. *BMC Bioinformatics* 8(472) (2007)
36. Swofford, D.L., Olsen, G.J., Waddell, P.J., Hillis, D.M.: *Phylogeny reconstruction*. In: *Molecular Systematics*, 2nd edn., ch. 11, pp. 411–501. Sinauer Associates, Inc., Sunderland (1996)
37. Van Laarhoven, P.J.M., Aarts, E.H.L.: *Simulated annealing: Theory and applications*. Kluwer Academic Publishers (1988)
38. Vazquez-Ortiz, K.E.: *Metaheurísticas para la resolución del problema de máxima parsimonia*. Master's thesis, LTI, Cinvestav - Tamaulipas, Cd. Vitoria, Tamps. Mexico (2011)
39. Waterman, M.S., Smith, T.F.: On the similarity of dendrograms. *Journal of Theoretical Biology* 73(4), 784–800 (1978)
40. Xiong, J.: *Essential Bioinformatics*, 1st edn. Cambridge University Press (2006)

**Part V**  
**Probabilistic Modeling and Optimization**  
**for Emerging Networks**

# A Bayesian Network Based Critical Infrastructure Risk Model

Thomas Schaberreiter, Pascal Bouvry, Juha Röning, and Djamel Khadraoui

**Abstract.** Critical infrastructures (CIs) provide important services to society and economy, like electricity, or communication networks to enable telephone calls and internet access. CI services are expected to provide safety and security features like data *Confidentiality* and *Integrity* as well as to ensure service *Availability* (CIA). The complexity and interdependency of CI services makes it hard for CI providers to guarantee those features or even to be able to monitor the CIA risk by taking into account that an incident in one CI service can cascade to another CI service due to a dependency.

CI security modelling tries to address some of the problems by providing a model for on-line risk monitoring. The model displays risk on the CI service level and can capture the dependencies to other CI services and include them in risk estimation. In this work Bayesian networks (BNs) are introduced to the CI security model to provide a method to derive CI service risk and allow features like risk prediction and handling of interdependencies. To the best of our knowledge this is the first time that BNs are used for on-line risk estimation in CIs.

## 1 Introduction

Critical infrastructure (CI) security has become an important research topic in the last years. CIs are service providers, the services they provide are so vital to the

---

Thomas Schaberreiter · Djamel Khadraoui  
Centre de Recherche Public Henri Tudor; Service Science  
& Innovation (SSI); 29, avenue John F. Kennedy, L-1855 Luxembourg, Luxembourg  
e-mail: [thomas.schaberreiter,djamel.khadraoui}@tudor.lu](mailto:{thomas.schaberreiter,djamel.khadraoui}@tudor.lu)

Pascal Bouvry  
University of Luxembourg; Computer Science and Communications Research Unit; 6, rue  
Richard Coudenhove-Kalergi, L-1359 Luxembourg  
e-mail: [pascal.bouvry@uni.lu](mailto:pascal.bouvry@uni.lu)

Juha Röning  
University of Oulu; Department of Electrical and Information Engineering; P.O.Box 4500,  
FIN-90014 University of Oulu, Finland  
e-mail: [juha.roning@ee.oulu.fi](mailto:juha.roning@ee.oulu.fi)

social and economic well-being of a society that a disruption or destruction of the infrastructure would have severe consequences. CI sectors include, amongst others, the telecommunication, electricity and transport infrastructures.

Security in CIs can be seen from many viewpoints. In this work, the term CI security (or CI risk) is used to define the risk of a breach of confidentiality, the risk of a breach of integrity and the risk of degrading availability (CIA) of a CI or CI service. During our research some key factors that influence CI security were identified: First of all, the *complexity* of CIs makes security complex. CIs are large organizations and their security is influenced by technical, social and organizational factors both at a national and international level. This complexity makes it hard to identify the most critical parts of a CI and their interactions which influence CI security. Secondly, the *dependencies* and *interdependencies* between CIs and CI services influence the security of CIs. CIs can be dependent on the service of other CIs and a security incident in one CI or CI service can cascade and cause an incident in another CI or CI service. Interdependencies exist when an incident cascades back to the initial service through dependent services. Thirdly, the *diversity* of CI sectors makes it hard to have a holistic view on CI security, e.g. by including dependencies to other CI sectors in CI risk estimation.

CI security modelling ([2], [1]) was introduced to address those problems and provide a CI model for on-line risk monitoring. The main entities of the CI security model are *CI services* and *dependencies* between CI services. The model can estimate the CI service risk after a security incident (observed by *system measurements*) and distribute this estimate to dependent CI services which in turn can use this information to update their risk estimate. The estimated CI service risk level is represented by a risk value between one (no risk) and five (maximum risk). The strong points of the CI security model are that by using CI service risk as the model output, uniform and comparable data is created that is valid for all CI sectors, and that dependencies can be included in CI service risk estimation.

In this work, for the first time, Bayesian networks (BNs) are introduced to the CI security model. BNs in this context provide a way to estimate risk from system measurements and dependent CI service risk, providing a more sophisticated way to model risk as well as some advanced features like *risk prediction* and handling of *interdependencies* using Dynamic Bayesian networks (DBNs).

## 2 Related Work

The concept of CI security modelling relates to several research areas: CI modelling and simulation, CI (inter)dependency identification and risk estimation in CIs. Identifying the various kinds of dependencies among CIs has been subject to previous research. In [1] Rinaldi et al. provide an excellent overview on the dimensions where interdependencies can occur. Several publications propose CI models based on various different modelling techniques. For example, conceptual modelling is used

in [15] by Sokolowski et al. to represent an abstract and simplified view of CIs. In [9] Panzieri et al. utilise the complex adaptive systems (CAS) approach for CI modelling. The model is derived by modelling the mutually dependent sub-systems of the infrastructure. Risk models for CIs were proposed by some authors. For example, in [8] Haslum et al. use continuous-time hidden Markov models for real-time risk calculation and estimation. In [3] Baiardi et al. propose a risk management strategy based on a hyper-graph model to detect complex attacks as well as to support risk mitigation. In [7] Haimes et al. propose an eight step risk ranking and filtering framework based on risk scenarios, using hierarchical holographic modelling. Bayes theorem is used to estimate the likelihood of risk scenarios. In general, previously published CI models and CI risk models vary greatly in their purpose and the extent to which they were implemented. The models are usually too high-level and therefore lack practical relevance or they are focused on a specific CI and therefore lack generality.

The idea of the CI security modelling differs greatly from the models previously published. It tries to establish abstract models of CIs that can be compared with each other while maintaining generality by enabling it to be applied to all kinds of CI sectors. CI security modelling is ongoing research, the first publications date from 2010 ([2], [1], [12]). The intent of those publications was to introduce the service based notion of CIs and to illustrate the dependence of services amongst another. The need of a common modelling entity to address the diversity of different CI sectors was introduced by using a risk-based CI model. Risk is estimated from system measurements using a weighted-sum method. One of the shortcomings of this original model is that it relies heavily on expert assessment. Despite evaluating the structure and dependencies of a CI, experts also assess the importance (weight) of each system measurement to a service in order to be able to calculate risk using a weighted sum. In this work BNs are introduced to the CI security model to provide a more sophisticated yet more convenient way of representing risk which allows to address shortcomings of the original proposal. Using this approach, system measurements do not need to be weighted for their importance which reduces the dependence on expert knowledge. Using BNs and their temporal extension, DBNs, as a modelling base also allows to include some features that were not covered by the original proposal, like risk prediction or a way to model interdependencies.

To address the complexity of CIs and to reduce the dependence on expert assessment in determining the structure of the CI security model, a CI analysis method based on dependency analysis was presented in [14]. The idea is to utilize different information sources on different organizational levels and to combine social as well as technical sources to get a holistic view on the investigated CI and to identify critical services, dependencies between critical services and system measurements to determine CI service states.

In [13] a set of indicators is presented that allow to evaluate the correctness of calculated service risk in form of an assurance indicator and in [4] the risk indicators received from dependent services are evaluated based on the estimated trust in those services.

### 3 Introduction to Bayesian Networks

BNs provide a way to model the probability of an event, given the state of events it depends on. BNs are directed acyclic graphs where the nodes represent event variables and the directed arcs represent the relationships or dependencies between the variables. Nodes in a BN are only dependent on their parent nodes, not on any other ancestors (conditional independence). In other words, conditional independence is given if a node is independent of its ancestors given the state of its parents. Each node in the BN has a conditional probability table (CPT) assigned to it containing the probabilities of the node being in a certain state, given the state of the parent nodes. Each possible combination of each state of the parent nodes has to be evaluated. It can be easily seen that the CPTs for nodes with a significant amount of parent nodes with multiple states are quite complex. The size of a CPT grows exponentially with the number of parent nodes.

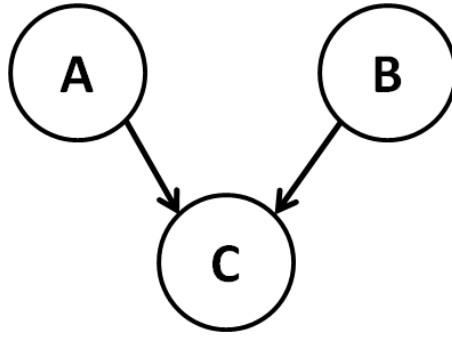
For illustrative reasons, Figure 1 represents a small abstract example of a BN. Let's assume three boolean variables  $A, B, C$ , where the state of  $C$  depends on  $A$  and  $B$ . In this simple example boolean variables are used, but BNs can handle multi-state or continuous variables as well. The aim of this simple model is to determine the state of the node  $C$  given the states of the nodes  $A$  and  $B$  ( $P(C | AB)$ ). The structure of the CPT for the variable  $C$  can be seen in Table 1. It contains all the possible combinations of the states of all involved nodes. The probabilities can either be learned from recorded data samples or assigned by an expert (assuming the expert has enough expertise and experience for the assessment). It can be seen that the sum of the probabilities over all the states of  $C$  has to be 1 for each combination of parent states.

The CPTs of variables without a parent (like variables  $A$  and  $B$ ) only contain the probabilities of the variable being in each state (marginal probabilities  $P(A)$  and  $P(B)$ ). Those probabilities represent how likely it is for node  $A$  or  $B$  to be in state "true" or "false".

**Table 1** Conditional probability table of variable  $C$

A		true		false	
B		true	false	true	false
C	true	p1	p2	p3	p4
	false	(1-p1)	(1-p2)	(1-p3)	(1-p4)

In this work the main objective of the BN is to be used as a classifier. In this case, one is interested in the most probable state of a variable given a combination of states of the parent variables. For example, it is observed that variable  $A$  is in state "true" and variable  $B$  is in state "false". What is the most probable state of variable  $C$ ?



**Fig. 1** Simple Bayesian network structure

In the context of modelling dependencies in CIs, BN classifiers provide a very natural and intuitive modelling approach.

Dynamic Bayesian networks (DBNs) are an extension of BNs to allow to model changing temporal relationships between variables. Both dependencies between variables and conditional probabilities can change over time. DBNs can model this by representing each time frame  $t$  by a separate BN and linking the time slices in the direction of the time flow.

## 4 BN Based Critical Infrastructure Security Model

Several reasons lead to using BNs for this model. First of all, the graphical structure of BNs is very intuitive and easy to interpret for people not familiar with the modelling domain, no model specific aspects need to be considered. Secondly, BNs have the capability of learning probabilities from data as well as being able to be assigned by experts. It is assumed that it is essential in the CI field, where sophisticated expert knowledge is available, experts will evaluate probabilities that can not be learned from data or re-evaluate probabilities that were learned from insufficient data. Furthermore, BNs provide a very natural way to model dependencies. One of the main goals of BNs is to capture the relationships and dependencies between events, as does the CI security model.

However, it is not easy to model complex interacting systems like CIs using BNs. The first problem is to find the right structure of the BN. The main difficulty is to tackle the complexity of CIs and to find the most critical systems and their dependencies. Also, having in mind the dependencies between CI sectors and the diversity of different CI sectors, it is hard to find a common ground for modelling. The CI model should output data that operators in all CI sectors can easily interpret and exchange. Using risk as a modelling objective provides a good common entity that concerns providers from each CI sector. The approach for finding the BN structure for CI security modelling will be described in Section [4.1](#).

Another aspect of the CI security model is to capture risk not only in the moment an incident happens (short-term effects), but also to model (or predict) the risk that this incident poses in the mid-term and long-term. In BNs this can be achieved using DBNs. This approach will be introduced in Section 4.3.

One of the shortcomings of using BN model for this work that it is not trivial to model directed cycles in BNs. Critical infrastructures can be interdependent. That means that an incident in CI A can cause an incident in CI B which in turn effects CI A. In a BN this represents a directed cycle. An approach to circumvent this problem using DBNs is presented in Section 4.4.

## 4.1 Structure for Bayesian Network

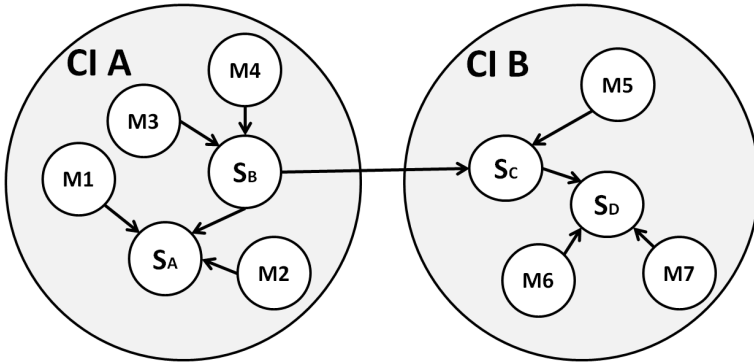
When building a model of complex systems, understanding the structure of the system and the interaction between components and other external systems is crucial. It is equally important to understand the purpose of the model to be able to map the real system to the abstract entities of the model.

In the CI security model the central modelling entities are *CI services* and the interactions (or *dependencies*) between services. A CI service is provided by a CI either to customers or to other CI services as a dependency. The main objective is to model CI service risk by observing *system measurements* that define the service state and by observing dependent CI service risk. The mapping between real-world observations and abstract risk estimates is done by the Bayesian classifier.

The main concern when building a CI model is complexity. Is it feasible to identify the critical services and dependencies that adequately represent the structure in a complex system like a CI and to represent them in a model? In an attempt to address this question a CI analysis method based on dependency analysis was proposed in previous work ([14]). This method adapts the PROTOSMATINE dependency analysis method ([6], [10]) to be used to find the structure for the CI security model and to identify the modelling entities used in the CI security model (critical services, dependencies and system measurements). The method is based on the assumption that in order to get a holistic view of a complex system, all available information sources (e.g. documentation, manuals, interviews, contracts, ...) on all levels in an organization (management, process, technical) need to be combined. The outcome of this method is a graph that contains the critical services as nodes and their dependencies as edges. System measurements that define a CI service state can be seen as dependencies of this service. An example for the structure of a CI security model can be seen in Figure 2. The nodes  $S_A$  to  $S_D$  represent critical services, the nodes  $M_1$  to  $M_7$  represent system measurements. The edges represent a dependency between services (e.g. the availability of  $S_A$  depends on  $S_B$ ) or between a service and a system measurement (e.g. the availability of  $S_A$  depends on the state of  $M_1$ ). The grey cycles



CI A and CI B do not have any purpose in the model, they should illustrate that dependent services can also belong to different critical infrastructures (e.g.  $S_B$  of CI A is needed to ensure availability of  $S_C$  in CI B).



**Fig. 2** Example CI security model structure

It is assumed that structure learning in this work is done using the above mentioned method. Automatic BN structure learning algorithms (like K2 presented in [5]) are available, but it is assumed that in the context of CIs it will be hard to capture the complex social, technical and organizational relationships using unsupervised structure learning. The system is not defined by a known set of variables where the problem is to find the optimal set of dependencies between them. In this context, the main problem is to identify an abstract set of services which might not be directly visible from observing data and to identify the relationships (or dependencies) between them. Structure learning is out of scope of this research, however it is not impossible that BN structure learning algorithms can be useful for this model, especially on the purely technical side of CIs where system behaviour can be determined exclusively from data records.

## 4.2 Conditional Probability Tables

After evaluating the structure of the BN, the next step is to evaluate the CPTs for each node. As mentioned before, the BN is used as a classifier and the main interest is in determining the most probable state of a node for each combination of states of the parent nodes.

The following list contains a summary of definitions and pre-conditions to be able to learn the conditional probability tables for the nodes in the context of the CI security model:

- (i) The state of a CI service node represents risk (for example, C, I or A). The risk value is limited to 5 discrete states, with 1 representing lowest risk and 5 representing highest risk.
- (ii) System measurement usually represent continuous measurements. In order to be more easily processed, they are pre-processed to 5 discrete values with 1 representing a measurement during normal system operation and 5 representing a maximum allowed deviation from normal operation. Boolean-type measurements will only have the states 1 (normal operation) and 5 (abnormal operation).
- (iii) System measurements need to be observable and it is assumed that their values can be recorded over time and can be combined with a time stamp.
- (iv) Each CPT represents the probabilities for one risk indicator (either C, I or A). If more than one risk indicator is interesting for a CI service, probabilities for each indicator have to be estimated separately.

For learning CPTs, 3 different types of nodes have to be distinguished: *System measurement nodes*, *CI service nodes without dependencies to other services* and *CI service nodes with dependencies to other services*.

System measurements can be seen as nodes without parents ( $M1 - M7$  in Figure 2), the CPT of those nodes will contain the probabilities of the system measurement being in a certain state, which can be easily learned from recorded data samples.

Learning nodes that have only base measurements as parents (services with no dependencies to other services, like  $S_B$  and  $S_C$  in Figure 2) is a matter of mapping the state combinations of the service node to a service risk level. Since the service risk is an abstract concept that can not be directly measured, an expert has to evaluate the risk a service experienced during the time period used to learn the probabilities (for example, an expert estimates that from time  $x$  to time  $y$  the service faced an incident that can be classified confidentiality risk level 3). With this information, the probabilities with respect to the states of the parent nodes can be learned from recorded data samples.

Learning nodes with dependencies to other services (like  $S_A$  and  $S_D$  in Figure 2) is similar to learning nodes without dependencies to other services, with the difference that the data used to learn the probabilities does not only come from system measurements, but also contains abstract service risk from dependent services. This does not make a difference for the BN since different types of information can be used to learn probabilities, but service risk estimates for the dependent services (like described in the previous paragraph) need to be available for the time period that is used to learn the probabilities.

Probabilities that can not be learned from data samples because of incomplete data (state combinations that never happened or only rarely happened) can be supplemented by experts. This can be a burden since CPTs might be complex (CPTs grow exponentially with the number of parents of a node). Since the BN in this work is used as classifier and only the most probable state of a parent state combination is of interest, this burden can be reduced. An interesting aspect in the validation phase of the model will be to evaluate the burden for a CI expert in the assessment

of a real-world CIs. For example, how many nodes and dependencies can be in the model so that an expert who has to validate the model still feels comfortable in providing input. Also, how complex can CPTs be so that an expert feels comfortable to validate and estimate probabilities in the CPT.

### 4.3 Risk Prediction

In the previous chapter the learned probabilities only capture the current risk of a service (if there is a certain combination of parent node states, what is the most probable risk of the service). In practice companies are usually also interested in the evolution of risk over time after an event occurred. One way of representing this is to estimate the *short-term* (e.g. hours after an event) *mid-term* (e.g. weeks after an event) and *long-term* (e.g. months after an event) effects of an event or incident. In the BN model this can be represented using DBNs. The basic idea is to separate the data that is used to learn the BN into time frames after an event happened. A CPT can be learned (or estimated by an expert) for each time frame. This will give an estimate of the most probable state of a service in each time frame, e.g. if an incident happens, what is the risk the service faces in the next hours, the next weeks and the next months.

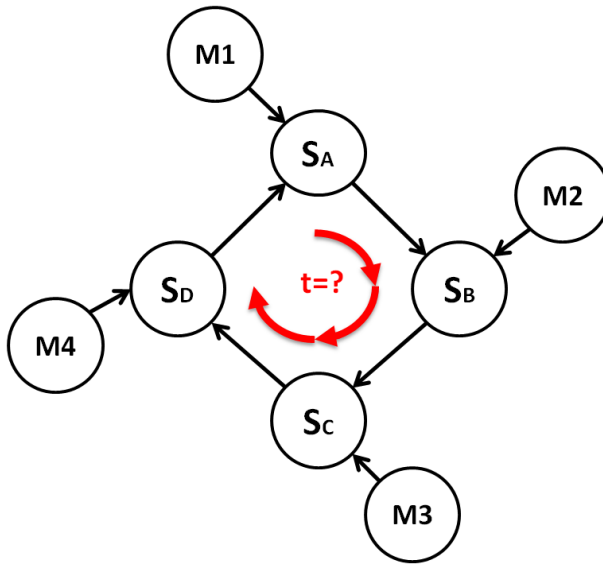
Using DBN for risk prediction makes learning the CPTs considerably more complex. More CPTs need to be considered since each node has a separate CPT in each time frame and the amount of data needed to learn grows in terms of the time that has to be considered as well as the amount of recorded incidents to be able to capture all possible states in each time frame. Furthermore, if a time frame of weeks or months is considered, it might not be easy to distinguish the effects of the original incident with effects of a possible later incident. CPTs for mid- and long-term prediction are assumed to be used primarily in services that represent the management and business level of a CI, not on services that represent the risk of day to day operations. It will depend on the CI the model is applied to, but the primary source of data for learning the CPTs for mid- and long term will not be recorded data samples, but expert knowledge.

### 4.4 Interdependencies - Directed Cycles in Bayesian Networks

One of the shortcomings of BN models is that it is not easily possible to model directed cycles. In the context of the CI security model this is a problem, since one of the main goals of the work is to model interdependencies between CI services. Interdependencies exist when an incident in one CI service effects the same service again through dependent services. An illustration of this behaviour can be seen in Figure 3 where the dependency cycle is  $S_A \rightarrow S_B \rightarrow S_C \rightarrow S_D \rightarrow S_A$ . The nodes  $M1, M2, M3, M4$  represent system measurements that can change the state of the service nodes.

In the BN this is represented by a directed cycle. The reason why the CPT probabilities can not be learned using classical BNs is that in a directed cycle the conditional independence condition is broken, nodes in the cycle do not only depend on the state of their parents, but also on other ancestors (in this particular case a node also depends on itself). In practice this means that it can not be distinguished if the effects of an incident originates from a CI service or secondary effects of a previous incident are experienced through a loop-back. The resulting probabilities for CI service risk might not reflect reality.

One way to address the shortcoming of BNs for this scenario is to use DBNs. The idea is to estimate the time ( $t$ ) it takes for an incident to loop back to a service through a dependent service. This time is taken as the time frame to build the DBN. The CPT in the first time frame represents the probabilities for CI service risk given an event without the loop-back effect, the CPT in the second time frame would represent the probabilities of service risk under the assumption that the effects of the original event loop back during this time frame. The CPTs of the third, fourth and  $n$ -th time frame represent the probabilities of service risk after the second, third and  $(n-1)$ th loop, respectively.



**Fig. 3** Example of cycle in BN

It is assumed that in the context of CI service risk, the main influences of the interdependency will be in the lower order loop-backs. This means that after an event increased the service risk and this increased risk looped back the first time, usually the influence of the interdependency will be covered already and higher order loop-backs will not result in a changed service risk.

As in the previous sections, the probabilities for the CPTs can be learned from data or they can be estimated by experts. Depending on the time interval of a loop,

it can be hard to learn the probabilities from records. An event that takes minutes or hours to loop back will be easier to learn than an event that takes months to loop back. Expert estimation will again be a necessary addition to estimate the probabilities that can not be learned from data.

## 5 Conclusions and Future Work

In this article a novel CI risk modelling approach for CI risk monitoring based on Bayesian networks was presented. The idea is to represent CIs as services they provide and the dependencies between the services. Risk is estimated based on observing system measurements that represent the CI service state as well as on observing risk of other CI services the service depends on. This model is represented as a BN where the nodes represent CI services and system measurements, and the edges represent dependencies between nodes. The probabilities in the CPT represent the most likely CI service risk, considering the state of the parent nodes (system measurements or CI services). Those probabilities can be learned from data records as well as being assigned by experts based on their experience with CI operation.

BNs were chosen for this model because some characteristics of BNs are seen as an advantage for the CI security model. The graphical structure of the BN is a direct representation of the modelled system which makes it easier for people not coming from the system modelling domain to visualize the resulting system model. Other advantages are that it is easy to combine data records and expert knowledge and BNs allow to combine evidence that represents diverse information which makes it easy to combine dependent system risk and system measurements to determine service risk.

Using a Bayesian approach for the CI security model allow some advanced features which were presented in this work. One feature is risk prediction. The idea is to estimate the risk not only right after an incident or service state change happened, but also estimate the mid- and long-term consequences of this event. Using DBNs it is possible to use BNs to estimate the effects of an incident in the short-term, mid-term and long-term. Another feature is the ability to model CI interdependencies. Effects of an incident in a CI service can loop back to the service through other services. This represents a directed cycle in BNs, which is not trivial to model. In this work DBNs are used to estimate the service risk with no loop-back effect, first order loop-back effect, ..., n-order loop-back effect.

Future work will focus on evaluation of the model in the context of a case study. The first part of the evaluation process will investigate the feasibility of CI analysis. The main question in this phase is if it is possible to extract critical CI services, system measurements and dependencies from complex systems. The second part of evaluation concerns CPT learning. The main questions in this phase will be if the BN approach will be manageable or if there will be too many evidence variables and therefore too complex CPTs, if the probabilities can be sufficiently learned from data and if expert estimation of probabilities is feasible. In the last phase of the validation the usefulness of a CI risk monitor will be evaluated with CI operators.

## References

1. Aubert, J., Schaberreiter, T., Incoul, C., Khadraoui, D.: Real-time security monitoring of interdependent services in critical infrastructures. case study of a risk-based approach. In: 21th European Safety and Reliability Conference, ESREL 2010 (2010)
2. Aubert, J., Schaberreiter, T., Incoul, C., Khadraoui, D., Gateau, B.: Risk-based methodology for real-time security monitoring of interdependent services in critical infrastructures. In: ARES 2010 International Conference on Availability, Reliability, and Security, pp. 262–267 (2010), doi:10.1109/ARES.2010.102
3. Baiardi, F., Telmon, C., Sgandurra, D.: Hierarchical, Model-based Risk Management of Critical Infrastructures. *Reliability Engineering & System Safety* 94(9), 1403–1415 (2009), doi:10.1016/j.res.2009.02.001; The 18th European Safety and Reliability Conference ESREL 2007
4. Caldeira, F., Schaberreiter, T., Monteiro, E., Aubert, J., Simoes, P., Khadraoui, D.: Trust based interdependency weighting for on-line risk monitoring in interdependent critical infrastructures. In: 2011 6th International Conference on Risk and Security of Internet and Systems, CRiSIS, pp. 1–7 (2011)
5. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.* 9(4), 309–347 (1992), <http://dx.doi.org/10.1023/A:1022649401552>, doi:10.1023/A:1022649401552
6. Eronen, J., Laakso, M.: A case for protocol dependency. *IEEE International Workshop on Critical Infrastructure Protection*, 22–32 (2005)
7. Haimes, Y.Y., Kaplan, S., Lambert, J.H.: Risk filtering, ranking, and management framework using hierarchical holographic modeling. In: *Risk Analysis*, vol. 22(2), Society for Risk Analysis (2002)
8. Haslum, K., Arnes, A.: Multisensor real-time risk assessment using continuous-time hidden markov models. In: 2006 International Conference on Computational Intelligence and Security, vol. 2, pp. 1536–1540 (2006), doi:10.1109/ICCIAS.2006.295318
9. Panziera, S., Setola, R., Ulivi, G.: An approach to model complex interdependent infrastructures. In: 16th IFAC World Congress (2005); CISIA, Critical Infrastructures
10. Pietikinen, P., Karjalainen, K., Eronen, J., Rning, J.: Socio-technical security assessment of a voip system. In: The Fourth International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2010 (2010)
11. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine* 21, 11–25 (2001)
12. Schaberreiter, T., Aubert, J., Khadraoui, D.: Critical infrastructure security modelling and rescu-monitor: A risk based critical infrastructure model. In: IST-Africa Conference Proceedings, pp. 1–8 (2011)
13. Schaberreiter, T., Caldeira, F., Aubert, J., Monteiro, E., Khadraoui, D., Simones, P.: Assurance and trust indicators to evaluate accuracy of on-line risk in critical infrastructures. In: 6th International Conference on Critical Information Infrastructure Security, CRITIS 2011 (2011)
14. Schaberreiter, T., Kittil, K., Halunen, K., Rning, J., Khadraoui, D.: Risk assessment in critical infrastructure security modelling based on dependency analysis (short paper). In: 6th International Conference on Critical Information Infrastructure Security, CRITIS 2011 (2011)
15. Sokolowski, J., Turnitsa, C., Diallo, S.: A conceptual modeling method for critical infrastructure modeling. In: 41st Annual Simulation Symposium, ANSS 2008, pp. 203–211 (2008), doi:10.1109/ANSS-41.2008.31

**Part VI**  
**Hybrid Probabilistic Models for Real**  
**Parameter Optimization and Their**  
**Applications**

# Adequate Variance Maintenance in a Normal EDA via the Potential-Selection Method

P.S. Ivvan Valdez, Arturo Hernández-Aguirre, and Salvador Botello

**Abstract.** It is a known issue that Estimation of Distribution Algorithms (EDAs) tend to converge prematurely. There exist various articles which report that they are incapable of determining the adequate variance, getting trapped even in regions which does not contain any global neither local minimum. In this vein, several proposals intend to insert and to preserve diversity. The proposal range is from modifying the probability distribution by inserting artificial variance, to mutating individuals or the search distribution. This work presents a novel selection method which is used to estimate a weighted covariance matrix equipped with adequate magnitude and directions. The covariance matrix is directly related with the information acquired from the current population, hence no artificial artifacts are inserted. By comparing with state of the art EDAs using test problems and graphical and statistical measures, we evidence that our proposal avoids premature convergence, and solves difficult problems -in the sense of variance preservation-, without the need of complex models, mutation, or explicit variance scaling.

## 1 Introduction

Estimation of Distribution Algorithms (EDAs) are based on estimating and sampling probability distributions. The aim is that the EDA, eventually, samples the optimum. The optimum is considered unknown, thus at the end of the search process we expect to have an accurate approximation to the optimum position as well as certain confidence about this approximation.

Indeed, from the point of view of the authors, the goals of an EDA are not only related with sampling the optimum, additionally, the search process must fulfill the following:

---

P.S. Ivvan Valdez, Arturo Hernández · Salvador Botello  
Centre or Research in Mathematics (CIMAT) A.C., C. Jalisco S/N,  
Mineral de Valenciana, Guanajuato, Gto. México. C.P. 36000  
e-mail: [ivvan, artha, botello@cimat.mx](mailto:ivvan, artha, botello@cimat.mx)



- (i) to sample intensively the most promising regions, assuming that one of such promising regions contains the optimum,
- (ii) to sample *all* the promising regions in the search space. Avoiding a promising region (a region which contain an attractor) could mean to avoid the region where the optimum is. This can be achieved by preserving and/or to computing adequate variance parameters, and,
- (iii) to converge to the elite individual. The EDA search process must lead the probability distribution to converge to the optimum, or at least to its best known approximation. Convergence to the best approximation intends to ensure the desired accuracy of the optimum estimator, by choosing the best approximation of several closest neighboring solutions. Furthermore, convergence provides a stopping criterion based on an exploration measure, instead of an arbitrary user given or expert given parameter such as the number of generations.

Our proposal aim is to fulfill these goals, they are discussed in the following subsections in order to support our approach.

### 1.1 Sampling Intensively the Most Promising Regions

An ideal EDA must sample with the greatest intensity the optimum region, and eventually converge to it. A probability function which behaves according inequality [\[1\]](#) could ensure the ideas just stated.

$$\int_X p(x,t)f(x)dx > \int_X p(x,t-1)f(x)dx. \quad (1)$$

Where  $f(x)$  is the objective function (for maximization), and  $p(x,t-1)$ ,  $p(x,t)$  are the search probability functions in any two consecutive generations. As  $f(x)$  does not change with  $t$ , thus  $p(x,t)$  must be increased around the maximum regions of  $f(x)$  and eventually must converge to the maximum. This idea can be simply stated as: *if the expected value of the objective function is increased each generation and a unique maximum exists in the search space, then the EDA with infinite population will converge to the optimum.* A deeper analysis and mathematical proofs of this argument can be found in [\[16\]](#).

Some theoretical probability functions can be used to tackle this problem, The characteristics needed to achieve this behavior for an ideal EDA are listed and contrasted with practical suggestions as follows:

- For **ideal EDAS** the better the objective function is in certain region, the greater the probability associated to it. For **practical EDAs** this could not be achieved for *all* the promising regions in the search space, due to the model of the parametric search distribution. On the other hand, we can easily set the maximum probability in *the containing region of the best approximation to the optimum we known (the elite individual).*

- In **ideal EDAs** the unique probability greater than 0 must be the corresponding to the optimum (considering an infinite sized population and unique optimum), for a large number of generations. On the other hand, to guarantee convergence for **practical EDAs**, the mean estimator of the objective values of the population, must be bounded by a non-decreasing sequence, consequently, if the objective function is bounded above, the finite sized population EDA will converge to the elite individual.

An ideal EDA which uses a distribution with the characteristics given above certainly, converges to the optimum [16]. The algorithm proposed in this article intends to follow the corresponding suggestions.

## 1.2 *Preserving and/or Computing Adequate Variance Parameters in Order to Maintain the Exploration Capacity*

The EDAs premature convergence problem has been a main subject of research. Since the early days of this field, researchers have noticed important lacks about variance and diversity preservation. By instance, studies determine that maximum likelihood base EDAs will never find the optimum unless the population grows exponentially [12]. In addition, researchers advise that maximum likelihood-based EDAs must integrate a variance scaling or a diversity preservation mechanism [7] in order to perform adequately. The proposals for improving EDAs exploration cover a wide range: from scaling the variance [2, 8], clustering [15], niching with diversity preservation mechanisms [5], to mutation [9]. Although they have been applied with different success ratio and cover different kinds of search spaces, they share some interesting characteristics:

- All of them state that maximum likelihood estimation is not the adequate method for computing the search distribution in EDAs.
- Most of them do not use directly the information into the population to determine the variance. Instead, they insert artificial information for increasing the variance such as mutation or crossover-like operators.
- All of them claim evident improvements when diversity is introduced in EDAs.
- The continuous cases work by analyzing and/or modifying the covariance matrix of a Normal (or Normal mixture) distribution.
- According to researchers, in continuous search spaces [7, 2, 8, 5, 15], the Rosenbrock function is the suitable test for this kind of proposals, and it can not be solved by continuous EDAs without variance scaling [2].

This work proposes a new selection-estimation method which captures enough information from the population which is sufficient to compute an adequate covariance matrix. The proposal is tested in (but no limited to) a continuous search space with a Normal search distribution model.

The article is presented as follows: this section introduces the main problem and ideas developed in it. Section 2 discusses about selection methods and the proposed

one which is the main subject of the paper. Section 3 presents an EDA based on the Normal distribution equipped with the proposed selection method.

Section 3 presents a set of experiments in order to contrast the novel proposal with state of the art algorithms, additionally, we present graphical and statistical analysis of the algorithm. Finally, Section 5 presents the main conclusions and perspectives of future work.

## 2 Selection Method

The goal of the selection method introduced in this section is two-fold: a) firstly, we ensure convergence to the elite individual, by using a truncation method which guarantees such convergence; b) secondly, we use a weighted selection method [14, 13], to insert the function landscape information and information about the coverage of the interest region into the parameter computation. The complete selection method can be seen as a weighted selection, where the selected set computed with the truncation method have a weight different than 0, and 0 otherwise. The truncation method which ensures converge to the elite individual, is presented in Algorithm 1.

---

**Algorithm 1** Truncation method to ensure increasing mean of the objective function and convergence to the elite individual. Maximization case.

---

**Input:**

$F$  vector of objective function values of size  $N$ ;

$\theta^t$  a threshold;

$\hat{I}^S \leftarrow \text{sort}(F, \text{decreasing}, \text{return\_index})$ ;

$\hat{I}^S \leftarrow \hat{I}_{1:(N/2)}^S$ ;

$I^S \leftarrow \hat{I}_i^S$  for all  $i \in \hat{I}^S$  such that  $F_{\hat{I}_i^S} \geq \theta$ ;

$M \leftarrow \text{sizeof}(I^S)$ ;

$\theta^{t+1} \leftarrow I_M^S$ ;

**Output:**

$I^S$  vector of indexes of selected individuals of size  $M$ ;

$\theta^{t+1}$  threshold for the next selection;

---

In Algorithm 1 the first threshold  $\theta^0$  is the worst objective value of the first generation, from the second generation in advance the others thresholds are computed inside the algorithm. Algorithm 1 ensures convergence and covering of the most promising solutions.

Notice that: a) if we have captured the optimal region, then, we just have to preserve it. b) All the individuals in the selected set are “good” individuals in the sense that all of them are better (in average) than those in the previous generation. c) If all individuals represent a promising region, the elite individual represents the best knowledge we have about the optimum position, but the farthest individual (to the elite) in the selected set is the most informative about the coverage of the region, and

also of the improvement direction. Hence, we can insert information or a *probability* in the covariance matrix estimator which indicates that the farthest individual to the elite is as important as the elite, this provides the adequate magnitude and direction in the covariance matrix to cover the interest region. For this purpose we propose a potential function as shown in Equation (2). It indicates that individuals are important because its objective function value as well as its distance to the optimum, considering that the farthest individual in the selected set is one of the most important ones for determining the promising region coverage. It is named a *potential function* recalling the functions used in physics for modeling forces in particle interactions, the forces among particles usually are related with the distances among them.

$$p(x_i) = \frac{\hat{p}(x_i)}{\sum_i \hat{p}(x_i)}, \text{ for } i \text{ in the selected set.} \quad (2)$$

Where:

$$\hat{p}(x_i) = \max(p_d(x_i), p_f(x_i)), \quad p_d(x_i) = \frac{\hat{p}_d(x_i) - \min(\hat{p}_d(x_i))}{\max(\hat{p}_d(x_i)) - \min(\hat{p}_d(x_i))}, \quad (3)$$

$$p_f(x_i) = \frac{f(x_i) - \min(f(x_i))}{\max(f(x_i)) - \min(f(x_i))}, \quad \text{and} \quad \hat{p}_d(x_i) = \sqrt{\sum_{j=1}^n (x_{i,j} - x_{best,j})^2}. \quad (4)$$

$f(x_i)$  is the objective function of the individual  $x_i$ , and  $n$  the number of variables. The potential function in Equation (2) returns the greatest value for the elite as well as for the farthest individual to the elite. This *normalized potential* can be used as probability, similarly to the probabilities used in [14], in order to compute a weighted estimator of the covariance matrix.

**A brief Comment on the Potential Function.** Potential functions are used in molecular mechanics to model interaction forces among molecules or particles (they are used also for proteins as particles). One of the applications is for simulating the transition of a configuration of particles to a stable state. Usually particle velocities and positions are computed via Newton formulae, using forces that become from different phenomena, one of such forces is given by a potential function which model the repulsion and attraction forces (Pauli repulsion and van der Waals attraction) among particles. Some cases, the potential function is truncated at a cut-off distance, considering that beyond such distance the effect of other particles is zero.

Even though our potential function is inspired in this phenomenon, the complete idea has a different meaning. Firstly the cut-off distance is given by the truncation selection, thus, particles (individuals) beyond such cut-off have a zero potential. Secondly, the potential function proposed here has the greatest values for the closest individuals to the elite in the objective function space (attraction in the objective function space), and the farthest individuals to the elite in the decision space (repulsion in the decision space). Thirdly, we intend to estimate a probability function

that fits the potential function instead of updating a particle configuration. In consequence, the desired density of particles (individuals) is proportional to the potential function. It is easy to say, we want to sample intensively individuals close to the elite in the function space and far away in the decision space.

The truncation and potential function are used to compute the covariance matrix in order to regulate the exploration. On the other hand, we use the mean estimator to ensure that the region around the most promising solution be the most intensively sampled, by simply setting the mean of the Normal distribution at the position of the elite individual, as stated in Equation (5).

$$\mu = x_{best} \quad (5)$$

The next Section introduces a Normal based EDA using the proposed potential function. We call our algorithm: Potential Selection based EDA (PS-EDA).

### 3 Potential Selection Based EDA

This section introduces a Normal EDA similar to those studied in [6]. All of them share the same basic framework, they use a multivariate Normal model. The proposal of this article is in Algorithm 2. In Algorithm 2,  $x_{best}$  is the elite individual. The covariance matrix computation is performed in lines 7 and 15. Notice that the proposal not only have similar cost of the covariance matrix computation than the standard approaches, furthermore, it avoids the use of complex models and operations. Another important advantage is that it only requires *one user-given parameter*, that is the population size. Even though it also requires a minimum norm of the covariance matrix as stopping criterion, it is considered input data related with the desired accuracy in the solution.

The example in Figure 1 shows the algorithm performance in a 2-dimension search space, the purpose is that the reader elucidates the behavior in higher dimensions. Figure 1 shows the Rosenbrock function (contour plot) and a population of 30 individuals asymmetrically initialized. In the first generation, the population is posed far away from the optimum, the elite individual is around the position  $(0, -1)$ , and the remaining population is in the left-bottom corner. Even though the population is confined to that corner, the potential forces the variance to cover the farthest selected individual to the elite, and, due to the symmetry of the Normal model, the variance, represented by the ellipse (level curve at 95% of probability), must be increased in the direction of the farthest individual to the optimum as well as to the opposite direction, resulting in a great gain in the exploration capacity of the algorithm. Notice that this way of computing the covariance matrix does not diminishes the convergence of the algorithm as is shown in Figure 1. As the covariance is upper bounded by the distance of the farthest individual in the selected set to the elite, it can not be arbitrarily enlarged.

**Algorithm 2** Normal multivariate EDA based on the Potential-Selection.**Input:****Problem parameters:**

$x_{inf}, x_{sup}$  vectors of inferior and superior limits respectively  
 $\epsilon_{tol}$  Minimum covariance matrix norm, for stopping criterion.  
 $n_{var}$  Number of variables

**User given parameters:**

$N$  Population size

```

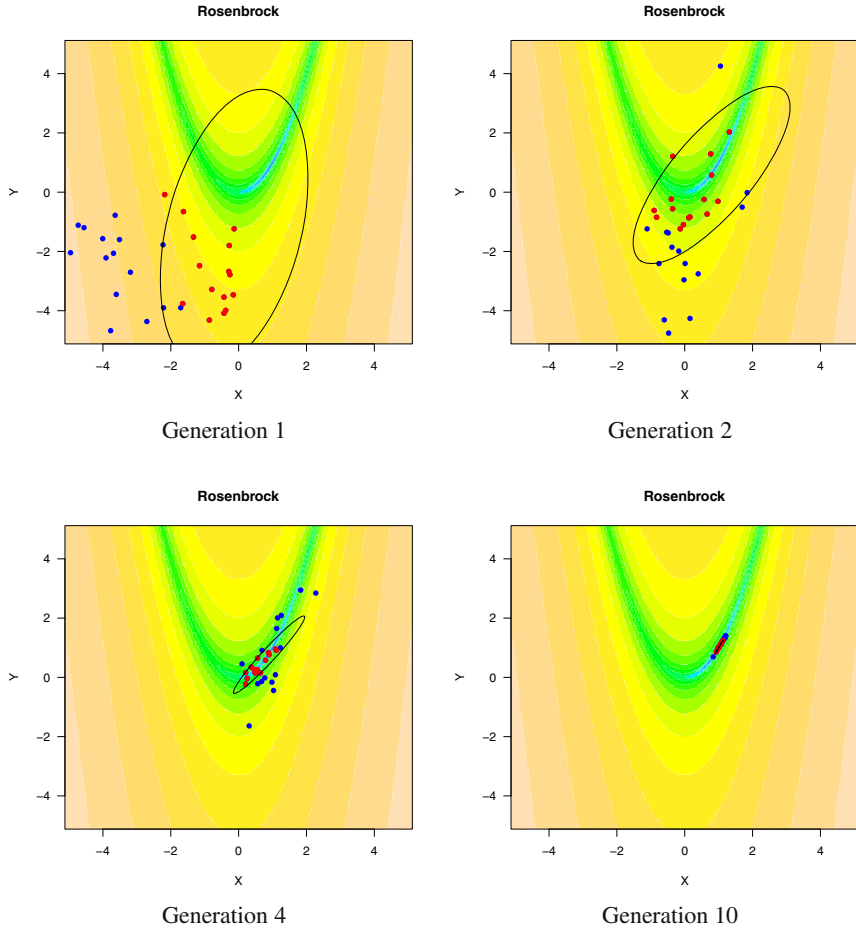
 $X \leftarrow \text{uniform}(N, x_{inf}, x_{sup});$ 
 $F \leftarrow \text{evaluation}(X);$ 
/* Truncation selection according Algorithm 1. */
 $I^S \leftarrow \text{selection}(F);$ 
/* Potential of the selected set, Eq. (2). */
 $p[I^S] \leftarrow \text{potential}(F[I^S]);$ 
 $\mu \leftarrow x_{best};$ 
/* Covariance matrix computation */
for  $i = 1..n_{var}$  do
    for  $j = 1..n_{var}$  do
         $\Sigma_{i,j} = \frac{\sum_{k \in I^S} (x_{k,i} - \mu_i)(x_{k,j} - \mu_j) p_k}{\sum_{k \in I^S} p_k}$ 
while  $|\Sigma| > \epsilon_{tol}$  do
     $X \leftarrow \text{Normal}(N, \Sigma, \mu, x_{inf}, x_{sup});$ 
     $F \leftarrow \text{evaluation}(X);$ 
     $I^S \leftarrow \text{selection}(F);$ 
     $p[I^S] \leftarrow \text{potential}(F[I^S]);$ 
     $\mu \leftarrow x_{best};$ 
    for  $i = 1..n_{var}$  do
        for  $j = 1..n_{var}$  do
             $\Sigma_{i,j} = \frac{\sum_{k \in I^S} (x_{k,i} - \mu_i)(x_{k,j} - \mu_j) p_k}{\sum_{k \in I^S} p_k}$ 
Output:
 $x_{best}$  Best optimum approximation.

```

## 4 Experiments and Results Discussion

In this Section we present a set of experiments to show the performance of the algorithm based on the potential-selection. The experiments use the test problems in Table 1.

The experiments are divided in subsections according to the results in the literature we are comparing with. For this comparisons we take into account several points: a) The comparing results are taken as they are reported in the literature, in order to avoid a bias of the comparison given by the implementation. In consequence, the parameters used for the algorithms we are comparing with can be



**Fig. 1** The potential-selection based Normal EDA for the Rosenbrock function

consulted in the corresponding reference. The parameter used in our approach is reported below for each particular experiment. b) The algorithms are compared with results reported numerically. Some researchers report graphical and qualitative results, and no sufficient numerical information is provided to replicate such graphs, hence, some graphical results of our proposal are presented for completeness, but they are not compared. In addition, in many cases, no sufficient statistics to perform hypothesis test are provided c) The comparisons are divided by article we are comparing with, due to the different conditions of the reported results, such as: stopping criteria, number of runs, measures reported, test problems, etc. We intend to perform a comparison as fair as possible following the criteria of the source article.

**Table 1** Test problems used in comparisons

Name	Definition	Domain
Rosenbrock	$\sum_{i=1}^{n-1} (100(x_i - x_{i+1}^2)^2 + (x_i - 1)^2)$	$x \in \{-5.19, 5.19\}^n$ for test 1 and 3 and $x \in \{-10, 5\}^n$ for test 2
Shifted Sum Cancellation	$f(x) = \frac{1}{10^{-5} + \sum_{i=1}^n  y_i }$ where $y_i = \sum_{k=1}^i (x_k - s)$	$x \in \{-0.16, 0.16\}^n$
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$x \in \{-10, 5\}^n$
Ellipsoid	$f(x) = \sum_{i=1}^n 10^{\frac{i-1}{n-1}} x_i^2$	$x \in \{-10, 5\}^n$
Cigar	$f(x) = x_1^2 + \sum_{i=2}^n 10^6 x_i^2$	$x \in \{-10, 5\}^n$
Tablet	$f(x) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$	$x \in \{-10, 5\}^n$
Cigar Tablet	$f(x) = x_1^2 + \sum_{i=2}^{n-1} 10^4 x_i^2 + 10^8 x_n^2$	$x \in \{-10, 5\}^n$
Two Axes	$f(x) = \sum_{i=1}^{n/2} 10^6 x_i^2 + \sum_{i=n/2+1}^n x_i^2$	$x \in \{-10, 5\}^n$
Different Powers	$f(x) = \sum_{i=1}^n 10^{2+10\frac{i-1}{n-1}}  x_i^2 $	$x \in \{-10, 5\}^n$

This section uses three articles of the specialized literature in order to compare the following:

- The first experiment compares the Potential-selection based EDA (PS-EDA) [4] versus algorithms that intend to capture sufficient fitness landscape information, through a complex structure of the search probability distribution. Thus, we compare our proposal which uses a single multivariate Normal distribution with several algorithms based on Gaussian mixtures. The purpose is to show that even a simple model with adequate position and variance could outperform complex and computationally-expensive models based on the classical EDA point of view of estimating best-fitted parameters from the selected set. Additionally, we shown that the PS-EDA can detect the promising regions, at least, with the same efficacy than Normal mixture models.
- The second experiment is a comparison with the first reported variance scaling scheme in multivariate Normal EDAs by Grahl et al. [8, 2]. This is one of the most impacting studies about diversity issues in Normal EDAs, and an obligated reference for the problem. The authors report coefficients of a log-log regression related to the computational cost and algorithm scaling. The purpose of this comparison is to show that our proposal can adequately solve and converge to the optimum with the desired precision ( $10^{-10}$ ), in spite of the high exploration capacity.
- The third experiment, is a comparison with another CVS scheme. Researchers notice that the CVS scheme proposed by Grahl fails to find the optimum when it is not positioned in the center of the search space (similar to asymmetrical initialization). Thus, our proposal is compared for this kind of problem, in effectiveness (optimum approximation) and efficiency (evaluations). With this experiment we shown that our approach solves a problem that the original CVS fails to solve, additionally, it is competitive with a CVS algorithm which solves the problem.



#### 4.1 Comparison with MIDEA, MBOA, and EDA with vbICA-MM

Cho and Zhang proposed an EDA with the variational Bayesian independent component analyzers mixture model named vbICA-MM [4]. The goal is to capture the structure of the data better than other simpler approaches. The algorithm was compared with the MIDEA [1] and MBOA [11] with the Rosenbrock function (the Fletcher-Powell function is also used but it uses random coefficients, thus the comparison could delivered inexact results). The Rosenbrock function has a small gradient and area of improvement close to the optimum, thus, many algorithms converge prematurely. We reproduce the comparison settings as follows:

**Comparison Settings:** 10 variables were used for the Rosenbrock function, and  $10^6$  function evaluations as stopping criterion. Mean and standard deviation of 20 independent runs are reported.

**Potential Selection Based EDA Settings:** Population size of 1000, the minimum norm of the covariance matrix (stopping criterion) was set in  $10e - 35$ .

**Table 2** Comparison among several EDAs in [4] for a 10-dimension Rosenbrock function

Algorithm	Mean and St. D. of $f(x_{best})$
<b>PS-EDA</b>	<b><math>0 \pm 0</math></b>
MBOA	$2.886 \times 10^{-2} \pm 0.0314$
EDA with vbICA-MM	$3.954 \pm 1.501$
MIDEA	$7.538 \pm 0.049$

**Results Discussion:** As can be notice in Table 2, our proposal clearly outperform other EDAs. In [4]  $10^6$  function evaluations were performed to obtain the cited results, in our case, the algorithm was stopped before by the minimum norm of the covariance matrix, thus the proposed algorithm does not perform  $10^6$  function evaluations, instead the mean and standard deviation of function evaluations for this problem were  $292100 \pm 7426.12$ , which which is around the 30% percent of the evaluations used by other algorithms. This experiment suggest that it does not matter how complex is the search distribution model, the variance reduces faster than required for an adequate exploration.

#### 4.2 Comparison with Adaptive Variance Scaling (Grahl et al.)

Grahl, Bosman and Rothlauf [8, 2] have presented two similar proposals for variance scaling in Normal EDAs named CVS-IDEA and SDR-CVS-IDEA. They enlarge the covariance matrix, after detecting the need of more variance. The functions used for testing the algorithms are not quite difficult to solve for Normal EDAs, because all of them are convex and most of them do not present high correlation among variables. On the other hand, some functions have interesting characteristics, several variables have a quite different impact than other in the objective function, then, most EDAs

usually tend to minimize the most important variables at first, and if the variance is reduced for all variables at the same time, they converge prematurely, which is the case of the Different Powers function. In order to solve this function the algorithm must maintain different convergence ratios or variances for the variables. As can be seen in the experiment, our algorithm successfully solve this kind of problem. The main comparison is performed with coefficients of a log-log regression. Such coefficients can be seen as an empirical order of the algorithms, they intend to measure how much the function evaluations grows related with the dimension of the problem. Nevertheless this kind of problem (convex with low correlation) is not the main niche of our proposal which is intended to solve harder ones. We present this comparison for completeness, and because the variance scaling proposal is consider one of the most promising and important proposals for solving the premature convergence issue in EDAs.  $\beta$  and  $\alpha$  are computed by fitting the models:  $\log(n_{eval}) = \beta \log(n_{dim}) + \beta_0 \log(n_{pop}) = \alpha \log(n_{dim}) + \alpha_0$ , for  $n_{pop}$  =population size,  $n_{eval}$  =number of evaluations and,  $n_{dim}$  =number of variables. Thus  $\beta$  is an exponent related to order of evaluation versus variables, and  $\alpha$  is an exponent related to the order of the population size versus the number of variables.

**Comparison Settings:** We compare the success of 30 independent runs in  $\{2, 4, 8, 10, 20, 40, 80\}$  dimensions for the following test problems: Sphere, Ellipsoid, Cigar, Tablet, Cigar Tablet, Two Axes, Different Powers and Rosenbrock. The Parabolic Ridge and Sharp Ridge reported in the original reference were not tested because they have the optimum in an infinity value, a rule for maintaining the population in the search space must be applied for this case, but setting and arbitrary criterion for this purpose could lead to an unfair comparison.

**Potential Selection Based EDA Settings:** For the dimensions  $\{2, 4, 8, 10, 20, 40, 80\}$ , the population sizes are:  $\{80, 120, 160, 160, 500, 2000, 8000\}$ , for all test problems except for the ellipsoid and Rosenbrock problems, for these problems the populations sizes are:  $\{80, 120, 160, 160, 500, 2000, 8500\}$ . The maximum norm of the covariance matrices are:  $\{1e-19, 1e-22, 1e-22, 1e-22, 1e-24, 1e-24, 1e-26\}$  for all problems except for the different powers, ellipsoid and Rosenbrock. For the ellipsoid and Rosenbrock they are:  $\{1e-19, 1e-22, 1e-22, 1e-22, 1e-24, 1e-24, 1e-27\}$ . For the different powers they are:  $\{1e-5, 1e-6, 1e-8, 1e-10, 1e-12, 1e-14, 1e-18\}$ .

**Results Discussion:** As noticed in Table 3 the PS-EDA has a greater empirical order than the other EDAs in the comparison, the positive fact is that it can solve all the problems in the test with the required accuracy. The  $\beta$  coefficient is an order related with the number of evaluation, and  $\alpha$  with the population size, the numbers reported show that the evaluations grows fater than the population size, this is important for memory aspects. As mentioned most of these problems are easy to solve for well performed EDAs, because their convexity, the next subsection will provide evidence about the PS-EDA in problems that the AVS-IDEA fails to solve.

**Table 3** Comparison on the empirical order for number of evaluations and population sizes. PS-EDA= Potential Selection based EDA.

Function	Algorithm	$\alpha$	$\beta$	Function	Algorithm	$\alpha$	$\beta$
Cigar	IDEA	0.5052	1.1865	Cigar Tablet	IDEA	0.4521	1.1142
	AVS-IDEA	0.2125	1.6976		AVS-IDEA	0.1879	1.7155
	CMA-ES	0.000	1.5183		CMA-ES	0.0000	1.2431
	PS-EDA	2.4171	1.2575		PS-EDA	2.3723	1.2575
Different Powers	IDEA	0.9355	1.4983	Ellipsoid	IDEA	0.6119	1.2171
	AVS-IDEA	0.8419	1.1692		AVS-IDEA	0.1870	1.6870
	CMA-ES	0.0000	1.5845		CMA-ES	0.0000	1.5183
	PS-EDA	2.6314	1.2575		PS-EDA	2.4158	1.26989
Rosenbrock	IDEA	not	solved	Sphere	IDEA	0.5541	1.1635
	AVS-IDEA	0.7475	1.9154		AVS-IDEA	0.1994	1.6563
	CMA-ES	0.6885	1.4872		CMA-ES	0.0000	0.9601
	PS-EDA	2.4032	1.26989		PS-EDA	2.4152	1.2575
Tablet	IDEA	0.4398	1.0860	Two Axes	IDEA	0.6603	1.2854
	AVS-IDEA	0.2066	1.6397		AVS-IDEA	0.2177	1.6551
	CMA-ES	0.0000	1.4178		CMA-ES	0.0000	1.7208
	PS-EDA	2.3476	1.2575		PS-EDA	2.4098	1.2575

### 4.3 Comparison with Adaptive Variance Scaling Schemes (Yungpeng et al.)

In [3], a failure in the adaptive variance scaling scheme [8] is detected, and a proposal to solve it is presented. They compare their proposal EMNA-CEAVS with the standard EMNA [10] and other variance scaling algorithm the EMNA-AVS [8]. The comparison is made by using the Shifted Sum Cancellation problem in Table 1. The reported results as well as the comparison with our proposal are show in Table 4. Notice that when  $s = 0$  in the Shifted Sum Cancellation, it is equal to the widely used Sum Cancellation Problem.

**Comparison Settings:** 10 variables were used for the Shifted Sum Cancellation function. The comparing algorithms were stopped when they found an optimum approximation with an error less than  $10^{-10}$  for 20 independent runs. The algorithm parameters were set in order of finding 95 % of successful runs and the average of functions evaluation is reported. When the algorithm does not solved successfully the problem, the mean of the error in the objective function space is reported.

**Potential Selection Based EDA Settings:** Population size of 300, the minimum norm of the covariance matrix (stopping criterion) was set in  $10e - 42$ , in order to find the desired precision.

**Results Discussion:** The results in Table 4 shows that our proposal is capable of solving problems which require high precision. Nevertheless the number of function evaluations are more than the best performed algorithm for several cases, our proposal does not seem to be conditioned by the  $s$  parameter of the objective function.

**Table 4** Comparison among several EDAs in [3] for a 10-dimension Shifted Sum Cancellation function. Mean of function evaluations when the optimum is approximated with an error less than  $10^{10}$ , otherwise the error mean is presented in brackets, for 20 independent runs. EDA-PS= EDA based on potential selection.

s	EMNA	EMNA-AVS	EMNA-CEAVS	EDA-PS
0	43020	45810	30930	$134745 \pm 2909.056$
0.01	172830	$[4e - 8]$	30930	$130455 \pm 6463.865$
0.04	185197	$[2e - 7]$	59507	$124890 \pm 3003.314$
0.16	280510	$[8e - 7]$	60872	$122970 \pm 4978.173$
0.64	$[1e - 6]$	$[3e - 6]$	64037	$118995 \pm 4339.698$
4	$[1 + 5]$	$[1.9e - 5]$	209000	<b><math>115920 \pm 5814.284</math></b>
64	$[1e5]$	$[3.8e - 4]$	206673	<b><math>118875 \pm 8419.643</math></b>

For all the experiments the PS-EDA found the optimum with the desired precision, and it worth to notice that the number of evaluations are quite similar for all the experiments in the table, while for the EMNA-CEAVS it is increased with the  $s$  parameters, The other algorithms in the comparison are not successful in solving the problem for all cases, thus the objective function mean and variance is reported. Recall that adaptive variance requires the computation of several statistical measures, while the potential calculation is quite simple.

## 5 Conclusions

This article presents a novel approach for continuous optimization using a Normal Estimation of Distribution Algorithm. Most (Normal) EDAs suffer of the premature convergence problem. The variance is reduced even if there is evidence about adequate search directions or regions. Our approach, called the Potential Selection, is a method to circumvent the mentioned issue. The potential selection computes weights which can be seen as *a priori* probabilities. Then, they are used to compute weighted estimator of the covariance matrix equipped with adequate magnitude and search direction. In this work we argue that the greatest probability must be positioned at the best known point. For the Normal distribution this is quite easy to achieve, because the maximum probability is defined by the means vector. The adequate pose of the probability mass improves the search capacity by sampling intensively the most promising regions, and the potential selection maintains the adequate variance. Additionally, convergence of the Normal EDA based on the potential selection is guaranteed, hence there is no need of an arbitrary stopping criterion. The covariance matrix norm can be used as stopping criterion according to the desired precision of the optimum approximation. The results are very promissory, the EDA is capable of solving all the test problems used in the comparisons which are taken from the specialized literature on EDAs with diversity enhancement methods.

As mentioned, it is desirable in EDAs to find an accurate optimum estimator as well as to have certain confidence about it. Our algorithm guarantees convergence

to the optimum estimator (elite individual), hence, the last populations are sampled in a close region around it, in consequence, implicitly we have evidence that our optimum estimator is the best found in a small neighborhood.

Our proposal intends to show that EDAs can be enhanced by simple procedures derived from conceptual work. In this proposal, we avoid the use of complex search distribution models, instead, we are focused on positioning the probability mass in the right region, and using the information in the population to determine the adequate direction and magnitude of the covariance matrix. We have shown that such information is actually present in the population. Also, we have provided arguments to elucidate that Estimation of Distribution Algorithms must not be focused on estimating an accurate distribution from data or to fit complex models via maximum likelihood. We conclude from our work and experiments that: a) EDAs must ensure that the regions with the highest objective values be intensively sampled, b) that all detected regions with a high objective function value must be explored, and finally c) must lead the population to converge around the best optimum estimator found.

Future work contemplate using the potential selection on discrete search spaces, and to investigate other potential functions which could outperform the approach presented here, possibly by reducing the number of function evaluations.

## References

1. Bosman, P.A., Thierens, D.: Advancing continuous ideas with mixture distributions and factorization selection metrics. In: Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO 2001, pp. 208–212. Morgan Kaufmann (2001)
2. Bosman, P.A.N., Grahl, J., Rothlauf, F.: SDR: A Better Trigger for Adaptive Variance Scaling in Normal EDAs. In: GECCO 2007: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 516–522. ACM (2007)
3. Cai, Y., Sun, X., Xu, H., Jia, P.: Cross entropy and adaptive variance scaling in continuous eda. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO 2007, pp. 609–616. ACM, New York (2007), doi:10.1145/1276958.1277081
4. Cho, D.-Y., Zhang, B.-T.: Evolutionary Continuous Optimization by Distribution Estimation with Variational Bayesian Independent Component Analyzers Mixture Model. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN VIII 2004. LNCS, vol. 3242, pp. 212–221. Springer, Heidelberg (2004)
5. Dong, W., Yao, X.: NichingEDA: Utilizing the diversity inside a population of EDAs for continuous optimization. In: IEEE Congress on Evolutionary Computation, pp. 1260–1267 (2008)
6. Dong, W., Yao, X.: Unified eigen analysis on multivariate gaussian based estimation of distribution algorithms. *Information Sciences* 178(15), 3000–3023 (2008), doi:10.1016/j.ins.2008.01.021
7. Grahl, J., Bosman, P.A.N., Minner, S.: Convergence Phases, Variance Trajectories, and Runtime Analysis of Continuous EDAs. In: GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 516–522. ACM (2007)

8. Grahl, J., Bosman, P.A.N., Rothlauf, F.: The correlation-triggered adaptive variance scaling idea. In: GECCO 2006: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 397–404. ACM Press (2006)
9. Handa, H.: The effectiveness of mutation operation in the case of estimation of distribution algorithms. *Biosystems* 87(23), 243–251 (2007)
10. Larrañaga, P., Lozano, J.A.: *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Norwell (2001)
11. Ocenasek, J., Schwarz, J.: Estimation of distribution algorithm for mixed continuous-discrete optimization problems. In: 2nd Euro-International Symposium on Computational Intelligence, pp. 227–232. IOS Press (2002)
12. Shapiro, J.L.: Diversity Loss in General Estimation of Distribution Algorithms. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 92–101. Springer, Heidelberg (2006)
13. Teytaud, F., Teytaud, O.: Why one must use reweighting in estimation of distribution algorithms. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO 2009, pp. 453–460. ACM, New York (2009),  
<http://doi.acm.org/10.1145/1569901.1569964>,  
doi: <http://doi.acm.org/10.1145/1569901.1569964>
14. Valdez, S.I., Hernández, A., Botello, S.: Efficient Estimation of Distribution Algorithms by Using the Empirical Selection Distribution. In: Korosec, P. (ed.) *New Achievements in Evolutionary Computation*. InTech (2010)
15. Yuan, B., Gallagher, M.: On the importance of diversity maintenance in estimation of distribution algorithms. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, GECCO 2005, pp. 719–726. ACM, New York (2005),  
<http://doi.acm.org/10.1145/1068009.1068129>,  
doi: <http://doi.acm.org/10.1145/1068009.1068129>
16. Zhang, Q., Mühlenbein, H.: On the Convergence of a Class of Estimation of Distribution Algorithms. *IEEE Transactions on Evolutionary Computation* 8(2), 127–136 (2004)

# Linkage Learning Using Graphical Markov Model Structure: An Experimental Study

Eunice Esther Ponce-de-Leon-Senti and Elva Diaz-Diaz

**Abstract.** Linkage learning to identify the interaction structure of an optimization problem helps the evolutionary algorithms to search the optimal solution. Learning the structure of a distribution representing the interactions of the optimization problem is equivalent to learning the variables of the problem linkage. The objective of this paper is to test the efficiency of an EDA that use Boltzmann selection and a cliqued Gibbs sampler (named Adaptive Extended Tree Cliqued - EDA (AETC-EDA)) to learn the linkage of the problem and generate samples. Some optimization problems difficult for the Genetic Algorithms are used to test the proposed algorithm. As results of the experiment is to emphasize that the difficulty of the optimization, as assessed by the number of evaluations, is proportional to the sizes of the cliques of the learned models, that in time, is proportional to the structure of the test problem.

## 1 Introduction

In genetics, linkage is the tendency for alleles of different genes to be passed together from one generation to the next [20]. For the evolutionary algorithms is interesting to detect linkage groups for the underlying structure of the optimization problem. If the linkage groups are not known in advance they must be detected during the algorithm execution. This question of linkage learning was proposed by Holland [8]. He noted that the complexity of the adaptive systems comes from the interactions of alleles that reflects the adaptation of the genotype to the environment. This adaptation requires most of the time a nonlinear structure that changes with the change of the simultaneous appearance of groups of variable alleles (epistasis - some phenotype appears only with an exact combination of alleles). One of

---

Eunice Esther Ponce-de-Leon · Elva Diaz-Diaz  
Autonomous University of Aguascalientes, Ave. Universidad 940,  
Colonia Ciudad Universitaria, CP 20131, Aguascalientes, Mexico  
e-mail: [eponce, ediazd}@correo.uaa.mx](mailto:{eponce, ediazd}@correo.uaa.mx)

the formal models proposed by Holland to reflect this property is the discrete probabilistic model defined over the space of all the possible combinations of alleles. Specifically a Discrete Graphical Markov Model (DGMM) lends his structure of interactions to represent the linkage. Learning the structure of the graphical model is equivalent to linkage learning. The DGMM are a type of hierarchic log linear probabilistic graphical models [11]

In the course of optimization the algorithm iteratively obtains the structure of a graphical model (linkage learning) and use this information to construct a cliqued Gibbs sampler. The evaluation of each solution generated by the cliqued Gibbs sampler and the new structure obtained at each step orients the search to the global optimum. The cliqued Gibbs sampler uses a Boltzmann selection procedure [7].

In this paper the algorithm is tested with three non-overlapping functions and one overlapping function. The overlapping function is the more challenging because not only has linkage whitening blocks, but the blocks are overlapping so, there exists linking between blocks. This additional difficulty is represented and manipulated by the cliques of the unrestricted graphical models employed.

Some of the frequently used estimation of distribution algorithms (EDAs) [14] are, the univariate marginal distribution algorithm (UMDA) [12], the bivariate marginal distribution algorithm (BMDA) [17], and the bayesian optimization algorithm [16]. In the mentioned papers a list of problems challenge the algorithms to learn the linkage. Some of the test problems used in this paper, were selected from these cited papers.

In this paper the power and sensibility of a linkage learning algorithm is analyzed. First its power to help the search of the solutions space and to find the best solution is tested with 4 test benchmark problems, and second, the sensibility to learn the structure of the graphical model that can describe the optimization function is assessed.

In section two some definitions and concept needed to describe the algorithm are given. In section three the algorithm pseudocodes are described and explained. In section 4, the test functions are described. In section 5, the experiments are described, the parameters explained and the results are presented. In section 6, the results are discussed and the conclusion presented.

## 2 Definitions and Concepts

The fundamental definitions and concepts needed to construct and explain the algorithms are detailed in order to make the paper self contained. More explanation about the AETC-EDA can be consulted in [18].

Let  $S = \{s_1, s_2, \dots, s_v\}$  be a set of sites and let  $\mathcal{G} = \{\mathcal{G}_s, s \in S\}$  be the neighborhood system for  $S$ , meaning it any collection of subset of  $S$  for which

- 1)  $s \notin \mathcal{G}_s$  and
- 2)  $s \in \mathcal{G}_r \Leftrightarrow r \in \mathcal{G}_s$ .

$\mathcal{G}_s$  is the set of neighbors of  $s$ , and the pair  $\{S, \mathcal{G}\}$  is a graph.



**Definition 1** A subset  $C \subset S$  is a **clique** if every pair of distinct sites in  $C$  are neighbors.  $\mathcal{C} = \{C\}$  denotes the set of cliques. Let  $X = \{x_s, s \in S\}$  denotes any family of random binary variables indexed by  $S$ . Let  $\Omega$  be the set of all possible values of  $X$ , that is,  $\Omega = \{w = (x_1, x_2, \dots, x_v) : x_i \in \{0, 1\}\}$  is the sample space of all possible realizations of  $X$ .

**Definition 2**  $X$  is a **Markov random field (MRF)** with respect to  $\mathcal{G}$  if

$$P(X = w) > 0 \text{ for all } w \in \Omega \quad (1)$$

and

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathcal{G}_s) \text{ for every } s \in S \text{ and } w \in \Omega \quad (2)$$

where  $X$  denotes the random variable and  $w$  denotes the values that this variable can take.

**Definition 3** A **Gibbs distribution relative to**  $\{S, \mathcal{G}\}$  is a probability measure  $\pi$  on  $\Omega$  with the following representation

$$\pi(w) = \frac{1}{Z} e^{-U(w)/T} \quad (3)$$

where  $Z$  and  $T$  are constants.  $U$  is called the energy function and has the form

$$U(w) = \sum_{c \in \mathcal{C}} V_c(w) \quad (4)$$

Each  $V_c$  is a function on  $\Omega$  that only depends on the coordinates  $x_s$  of  $w$  for which  $s \in C$ , and  $C \subset S$ .

**Definition 4** The family  $\{V_c, c \in \mathcal{C}\}$  is called a **potential** and the constant

$$Z = \sum_w e^{-U(w)/T} \quad (5)$$

is called **the partition function**. The constant  $T$  is named temperature and it controls the degree of "peaking" in the density  $\pi$ .

**Theorem 1** (Equivalence theorem) Let  $\mathcal{G}$  be a neighborhood system. Then  $X$  is a MRF with respect to  $\mathcal{G}$  if and only if  $\pi(w) = P(X = w)$  is a Gibbs distribution with respect to  $\mathcal{G}$ .

A more extensive treatment can be seen in [7], [3], [9].

**Definition 5** **Gibbs Sampling** is a Markovian updating scheme that works as follows. Given an arbitrary starting set of values  $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_v^{(0)}) \in \Omega$ ,

one value  $x_i^{(0)}$  is drawn from the conditional distribution  $P(x_i|x_1^{(0)}, \dots, x_{i-1}^{(0)}, x_{i+1}^{(0)}, \dots, x_v^{(0)})$  for each  $i = 1, \dots, v$ , where  $v$  is the number of variables. So, each variable is visited in the natural order until  $v$ . After that a new individual  $x^{(1)} = (x_1^{(1)}, x_2^{(1)}, \dots, x_v^{(1)})$  is obtained.

Geman and Geman [7] demonstrated that for

$$t \rightarrow \infty, \quad x^{(t)} \rightarrow x$$

where  $x = (x_1, x_2, \dots, x_v)$  and  $t$  is the parameter of the process (if the process is an algorithm,  $t$  is an iteration). This sampling schema required  $v$  random variate generations, one for each state  $i$  of the schema.

As a component of the cliqued Gibbs sampler, the generator part, a selection procedure is used.

**Definition 6** Let  $x_i$  be a solution and  $f(x_i)$  be the function of  $x_i$  to optimize, then the **Boltzmann selection** procedure evaluates a solution  $x_i$

$$P(x_i) = \frac{\exp(-f(x_i))}{E(x_m)} \tag{6}$$

where  $E(x_m) = \sum_j f(x_j)p(x_j)$  is the expected value of all the elements in the current population.

$P(x_i)$  is not invariant to scaling but is invariant to translation [5].

Given the state  $i$  of the schema  $x^{(t)}$ , for the value  $P[x_i|x_1^{(t)}, \dots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \dots, x_v^{(t)}]$  a value for  $x_i^{(t+1)}$  is obtained selecting as follows: if  $p(x_i^{(t+1)}) > p(x_i^{(t)})$  select  $x_i^{(t+1)}$ , else calculate

$$q = \frac{p(x_i^{(t+1)})}{p(x_i^{(t)})}, \tag{7}$$

and if a random number uniformly generated in the interval  $[0, 1]$  is less than  $q$ , select  $x_i^{(t+1)}$ .

**Definition 7** The **K-L divergence from the probability model  $M$  to the data  $x$**  is given by the Kullback-Leibler information [11]

$$G^2(M, x) = \log(L(\hat{m}_n^M(x))) = -2 \sum_{i=1}^k x_i \log_2\left(\frac{\hat{m}_i^M}{x_i}\right). \tag{8}$$

where  $k$  is the sample number of different individuals,  $n$  is the total number of individuals, and  $\hat{m}_n^M$  is the maximum-likelihood parameter estimator of  $m_n^M$ .

The K-L divergence is also known as relative entropy, and can be interpreted as the amount of information in the sample  $x$  not explained by the model  $M$ , or the deviation from the model  $M$  to the data  $x$ . This K-L divergence is used to calculate **SMCI** (Definition 9) and in time **EMUBI** (Definition 10) in the next paragraphs.

**Definition 8** The *mutual information measure*  $I_{X_i, X_j}$  for all  $X_i, X_j \in X$  is defined as

$$I_{X_i, X_j} = I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}. \quad (9)$$

where  $P(x_i, x_j) = P(X_i = x_i, X_j = x_j)$ .

As a part of a strategy to learn a graphical Markov model, a statistical model complexity index (SMCI) is defined and tested by Diaz et al. [6]. Based on this index it is possible to obtain an evaluation of the sample complexity and to prognose the graphical model to explain the information contained in the sample.

The model representing the uniform distribution is denoted by  $M_0$ , and the model represented by a tree is denoted by  $M_T$ . If a sample is generated by a model  $M$  containing more edges than a tree, the information about the model  $M$  contained in this sample and not explained, when the model structure is approximated by a tree, may be assessed by the index defined as follows.

**Definition 9** Let  $x$  be a sample generated by a model  $M$ . The *statistical model complexity index (SMCI) of the model  $M$*  [6] is defined by the quantitative expression

$$SMCI(M, x | M_T) = \frac{G^2(M_T, x) - G^2(M, x)}{G^2(M_0, x)}. \quad (10)$$

This index can be named sample complexity index, because it is assessed by the quantity of information contained in the sample generated by the model  $M$ .

**Definition 10** Let  $G$  be the graph of the model  $M$ , and let  $v$  be the number of vertices, let  $MNE(v)$  be the maximum number of edges formed with  $v$  vertices. The *edge missing upper bound index (EMUBI)* (see [6]) is defined by

$$EMUBI_\tau(M, x | M_T) = \tau(MNE(v) - v + 1)SMCI(M, x | M_T). \quad (11)$$

where  $\tau$  is the window allowing a proportion of variability in the sample to get into the model [6]. This coefficient is a filter that allows the sample relevant information for the model structure construction.

This index is used to prognose the number of edges to add to a tree in order to approximate the complexity of the sample using the graphical model, in this case it is an unrestricted graphical Markov model.

The next two definitions are used to assess the similarity of the linkage structure learned by the algorithm and the linkage structure of the optimization problem.

**Definition 11** Given the learned graph  $G_1$ , and the generator graph  $G_2$ , a *graph similarity index of  $G_1$  respect to  $G_2$* ,  $GSI(G_1, G_2)$  is given by the number of common edges divided by the number of edges in the generator graph  $G_2$  [6]. Denote by  $E_1$  the set of edges from  $G_1$  and by  $E_2$  the set of edges from  $G_2$ , then the similarity index is given by:

$$GSI(G_1, G_2) = \frac{|E_1 \cap E_2|}{|E_2|}, \quad (12)$$

where  $|C|$  denotes the number of elements in the set  $C$ .

**Definition 12** Let  $\mathcal{L} = \{l_i\}$  the learned graph, and  $\mathcal{G} = \{g_j\}$  the generator graph given by their cliques  $l_i$  and  $g_j$  respectively

- a. If for all  $j$  there exists  $i$  such that  $l_i \supset g_j$  then it is said that  $\mathcal{L}$  **overlearns**  $\mathcal{G}$ .
- b. If for some (but not all)  $j$  there exists  $i$  such that  $l_i \supset g_j$  then it is said that  $\mathcal{L}$  **partially learns**  $\mathcal{G}$ , and
- c. The proportion of cliques learned is the number of  $g_j$  cliques contained in some  $l_i$  cliques, divided by  $|\mathcal{G}|$ , is named **index of learned cliques**.

The generator graph in this paper is the graph given by the variable blocks of the objective function. To calculate the index of learned cliques the following pseudocode is used.

---

**Algorithm 1** Index of learned cliques algorithm

---

**Input:**  $\mathcal{L}$  and  $\mathcal{G}$   
 Calculate  $|\mathcal{L}|$  and  $|\mathcal{G}|$   
 count  $\leftarrow$  0  
 Learning index  $\leftarrow$  0  
**for**  $i = 1$  to  $|\mathcal{G}|$  **do**  
   **for**  $j = 1$  to  $|\mathcal{L}|$  **do**  
     **if**  $g_j \subset l_i$  **then**  
       count  $\leftarrow$  count + 1  
**Output:** Learning index  $\leftarrow \frac{\text{count}}{|\mathcal{G}|}$

---

To apply the learning index it is necessary to note that a graphical model is constructed over hypergraphs [2] that are formed by subsets of a given set. Graphical models are a particular type of hypergraphs whose maximal subsets correspond to cliques of a graph. So, it makes sense to compare cliques as subsets.

### 3 The Adaptive Extended Tree Cliqued - EDA (AETC-EDA)

The CL algorithm (Chow and Liu algorithm) [4] obtains the maximum weight spanning tree using the Kruskal algorithm [10] and the mutual information values  $I_{X_i X_j}$  (Definition 8) for the random variables. The tree obtained by this algorithm is denoted by  $M_T(CL)$ .

The CL-algorithm calculates for the number of variables  $v$ ,  $\frac{v(v-1)}{2}$  mutual informations. If each variable takes two values then the CL-algorithm has complexity

**Algorithm 2** CL algorithm

**Input:** Distribution  $P$  over the random vector  $X = (X_1, X_2, \dots, X_v)$   
 Compute marginal distributions  $P_{X_i}, P_{X_i, X_j}$ , for all  $X_i, X_j \in X$ .  
 Compute mutual information values (Definition 8)  $I_{X_i, X_j}$  for all  $X_i, X_j \in X$ .  
 Order the values from high to low (w.r.t.) mutual information.  
 Obtain the maximum weight spanning tree  $M_T(CL)$  by the Kruskal algorithm [10].  
**Output:** The maximum weight spanning tree  $M_T(CL)$

**Algorithm 3** The extended tree adaptive learning algorithm (ETreeAL)

**Input:** Distribution  $P$  over the random vector  $X = (X_1, X_2, \dots, X_v)$   
 Call CL Algorithm in order to obtain the maximum spanning tree using the mutual information measure  $I_{X_i, X_j}$  (Definition 8) as weight edges.  
 Calculate the edge missing upper bound prediction index ( $EMUBI_\tau(M, x | M_T(CL))$ ).  
 Add to  $M_T(CL)$ ,  $\tau$  percent from missing edges in the order of the mutual information values (Definition 8).  
**Output:** Extended tree model structure  $M_{EXT}(CL)$

$O(\frac{v(v-1)}{2}2^2) = O(2v(v-1)) = O(v^2)$ . The ETreeAL adds the number of prognosed edges to the tree to obtain the model structure necessary to the CG-sampler input, and these operations have polynomial complexity.

This algorithm needs the cliques of the graphical models as input to fulfill the condition of equivalence required by theorem 1 (see Section 2). The variables of each clique are used together by the CG- Gibbs sampler optimizer to generate new individuals for the sample. Other authors used blocked Gibbs sampler [19], and that is why the name "Cliques Gibbs Sampler" was given to this sampler. Let  $f(x)$  be the objective function.  $\hat{E}(x_m)$  is the  $f(x)$  mean estimator. Without losing generality in the description of the CG-Sampler algorithm, we assume that the optimization problem is to obtain a minimum and a convenient change can be made to obtain a maximum.

Observing the Algorithm 14 it is seen that the algorithm input consists of a population of selected solutions, let  $P$  be the population and let the structure of the graphical model adjusted be given by its cliques,  $\mathcal{C} = \{c_1, \dots, c_k\}$ . In the outer cycle the iterations run through all elements of the population ( $N$ ). The first inner **for** runs through the number of cliques ( $k$ ) and the step 7 generates a marginal table for each clique and calculates a roulette selection running over the marginal table corresponding to the clique  $c_j$  for all  $j=1, \dots, k$ . Let  $CM = \max_{c_j \in \mathcal{C}} |c_j|$  then the worst case is when  $|c_j| = CM$  for all  $j$ . In this case the complexity is given by  $O((N)(k)2^{CM})$ , so the complexity of the CG-Sampler is exponential in the size of the maximum clique of the model.

---

**Algorithm 4** CG–Sampler Optimizer Algorithm
 

---

**Input:** A population of selected solutions, and the structure of the graphical model, given by its cliques

$P_{New} \leftarrow \{\}$

$i \leftarrow 1$

**repeat**

    Take the solution  $i$  of the selected solutions

**for**  $j = 1$  to Number of cliques **do**

        Take the clique  $j$

        Generate the marginal table of the clique  $j$

        Roulette Selection: Select the individual corresponding to the  $k$  cell of the marginal table according to its selection probability (Boltzmann Selection)

        Definition 6

**if**  $\hat{E}(x_m) < f(k)$  **then**

$P_{New} \leftarrow P_{New} \cup \{k\}$

**else**

        /\* Metropolis Step \*/

**if**  $Random \leq \frac{f(k)}{\hat{E}(x_m)}$  **then**

$P_{New} \leftarrow P_{New} \cup \{k\}$

**else**

$P_{New} \leftarrow P_{New} \cup \{i\}$

**if**  $i = (Percent\ of\ selection) * N / 100$  **then**

$i = 0$

**else**

$i = i + 1$

**until** (a new population  $P_{New}$  with size  $N$  is obtained);

**Output:** A new population  $P_{New}$  (Gibbsian Population)

---

### 3.1 Adaptive Extended Tree Cliqued – EDA (AETC–EDA) Pseudocode

The AETC - EDA employs the ETreeAl to obtain the Markov model structure of the population of solutions for each algorithm's iteration, with this structure the CG- sampler optimizer obtains the next population.

## 4 Test Functions

Four functions are employed to analyze the linkage problem in AETC- EDA. The selected functions are a sample from functions of different difficulties. All of them are deceptive functions proposed to study the genetic algorithm performance. The Overlapping  $Trap_5$  is the most difficult in this analysis.

**Algorithm 5** Adaptive Extended Tree Cliqued – EDA (AETC–EDA)

**Input:** Number of variables, function to optimize, population size, percent of selected individuals, stop criterion

Create the initial population of size  $N$  at random.

**repeat**

    Evaluate the population.

    Order the population and select a portion.

    With the selected portion of the population call the *ETreeAl* algorithm (Algorithm 3).

    Call cliqued Gibbs sampler (CG–Sampler) optimizer (Algorithm 14)

**until** (*Some stop criterion is met*);

**Output:** Solution of the optimization problem

In all functions we use  $v$  as the number of the variables,  $x_i$  is a binary variable for every  $i$ , and  $u = \sum x_i$  is the number of ones in the solution  $\mathbf{x} = (x_1, \dots, x_v)$ .

**F<sub>c2</sub> Deceptive Problem.** Proposed in [13] its auxiliary function and deceptive decomposable function are as follows.

$$f_{Muhl}^5 = \begin{cases} 3.0 & \text{for } x = (0, 0, 0, 0, 1) \\ 2.0 & \text{for } x = (0, 0, 0, 1, 1) \\ 1.0 & \text{for } x = (0, 0, 1, 1, 1) \\ 3.5 & \text{for } x = (1, 1, 1, 1, 1) \\ 4.0 & \text{for } x = (0, 0, 0, 0, 0) \\ 0.0 & \text{otherwise} \end{cases}$$

$$f_{c_2}(\mathbf{x}) = \sum_{i=1}^{\frac{v}{5}} f_{Muhl}^5(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i}) \quad (13)$$

**F<sub>3</sub> Deceptive Problem.** This problem has been proposed in [13]. Its auxiliary function and deceptive decomposable function are as follows.

$$f_{dec}^3 = \begin{cases} 2 & \text{for } u = 0 \\ 1 & \text{for } u = 1 \\ 0 & \text{for } u = 2 \\ 3 & \text{for } u = 3 \end{cases}$$

$$f_{3deceptive}(\mathbf{x}) = \sum_{i=1}^{\frac{v}{3}} f_{dec}^3(x_{3i-2}, x_{3i-1}, x_{3i}) \quad (14)$$

**Trap<sub>k</sub> Problem.** A Trap function of order  $k$  [16] can be defined as

$$Trap_k(u) = \begin{cases} k & u = k \\ k - 1 - u, & \text{otherwise} \end{cases}$$

$$f_{Trap_k}(\mathbf{x}) = \sum_{i=1}^{\frac{v}{k}} Trap_k(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i}) \quad (15)$$

We use  $k = 5$ .

**OverlappingTrap<sub>5</sub> Problem.** A Overlapping Trap function of order 5 [15] can be defined as

$$\begin{aligned} a_i &= Trap_5(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i}) \\ b_i &= Trap_5(x_{5i+1}, x_{5i+2}, x_{5i+3}, x_{5i+4}, x_{5i+5}) \\ a_{v/5} &= Trap_5(x_{5(v/5)-4}, x_{5(v/5)-3}, x_{5(v/5)-2}, x_{5(v/5)-1}, x_{5(v/5)}) \\ b_0 &= Trap_5(x_1, x_2, x_3, x_4, x_5) \end{aligned}$$

$$f_{OverlappingTrap_5}(\mathbf{x}) = \sum_{i=1}^{\frac{v}{5}-1} [a_i + \omega \phi(a_i + b_i)] + a_{v/5} + \omega \phi(a_{v/5} + b_0) \quad (16)$$

where  $\omega = 1$  and  $\phi$  is defined as

$$\phi(a + b) = \begin{cases} -1 & \text{if } (a + b) \equiv 0 \pmod{2} \\ +1 & \text{if } (a + b) \equiv 1 \pmod{2} \end{cases}$$

The optimal solutions are constructed with blocks of size 5. There are two different optimal solutions  $\mathbf{x}_1$  and  $\mathbf{x}_2$ .  $\mathbf{x}_1 = (000001111100000\dots0000011111)$  and  $\mathbf{x}_2 = (111110000011111\dots1111100000)$ , where  $v$  (number of variables) fulfills with  $v \equiv 0 \pmod{5}$ .

For the  $F_{c_2}$  function, the graphical model assumed had 10 cliques, the number of edges was 100, and all cliques had a size 5. For the  $F_3$  Deceptive function, the graphical model assumed had 16 cliques, the number of edges was 48, and the size of the cliques was 3. The  $Trap_5$  had the same graphical model as  $F_{c_2}$  function. In the case of  $OverlappingTrap_5$  the graphical model was more difficult to describe in terms of number of cliques, size of cliques and number of edges.

## 5 Experimental Results

The experiments reported were performed with the four test problems described in the section 4. The parameters are described in Table 1.

Some edges of the tree learned by the AETC-EDA in the first step were erroneous because of the structure of the function, that is, the blocks of the optimization function do not intersect. This event occurs in functions  $F_{c_2}$ ,  $F_3$  Deceptive, and  $Trap_5$ .

The program of the AETC-EDA had a memory constraint for the size of cliques. The maximum size permitted of a clique was 10.

The behavior of  $F_{c_2}$  function is better with population size 70 than size 90 as you can see in the tables 2 and 3 respect to Evaluations Number. The \* means in the case of the  $OverlappingTrap_5$  function, that the learned and similarity index was



**Table 1** AETC-EDA parameters description

Parameters	Description
Number of Evaluations	Mean number of evaluation in 30 runs
Number of Cliques	Mean Number of cliques of learned graph in 30 runs
Size of Cliques	Mean size of cliques of learned graph in 30 runs
Edge number	Mean number of edges of learned graph in 30 runs
Learning Index	Mean rate in 30 runs, of cliques of the generator graph contained in some one clique of the learned graph, divided by cliques number of generator graph (See Definition 12)
Similarity Index	Mean Similarity in 30 runs, of the learned graph respect to the generator graph (See Definition 11)
Mean best value	The mean best value obtained by the AETC-EDA in 30 runs
Sample Complexity $x$	$x$ be the complexity of a Population of AETC-EDA in each generation and run (See Definition 9). The $x$ is bounded by the value after the sign $<$

**Table 2** AETC – EDA (CG - Sampler with Boltzmann Selection and without Metropolis step). Percent of Selection= 60.  $\tau = 1$ . Number of Generations= 500.

Parameters	$F_{c_2}$	$F_3$ Deceptive	$Trap_5$	$OverlappingTrap_5$
Variables	50	48	50	50
Population size	70	70	70	70
Number of Evaluations	394.07	20413.8	512235.2	621473.8
Number of Cliques	37.87	50.97	39.2	286
Size of Cliques	2.45	3.03	3.68	4.42
Edge number	73.2	125.23	150.23	283.2
Learning Index	0.16	0.96	0.66	0.77*
Similarity Index	0.58	0.99	0.91	0.91*
The Optimum value	40	48	50	100
Mean best value	40	48	49.5	62
Sample Complexity $x$	$x < 0.043$	$x < 0.12$	$x < 0.15$	$x < 0.27$

**Table 3** AETC – EDA (CG - Sampler with Boltzmann Selection and without Metropolis step). Percent of Selection= 60.  $\tau = 1$ . Number of Generations= 500.

Parameters	$F_{c_2}$	$F_3$ Deceptive	$Trap_5$	$OverlappingTrap_5$
Variables	50	48	50	50
Population size	90	90	90	90
Number of Evaluations	656.4	28315.7	1122637.33	658040.4
Number of Cliques	39.57	43.2	30.53	85.9
Size of Cliques	2.35	2.86	3.6	4.4
Edge number	68.4	102.2	124.06	265.1
Learning Index	0.13	0.98	0.66	0.77*
Similarity Index	0.58	0.99	0.91	0.92*
The Optimum value	40	48	50	100
Mean best value	40	48	49.46	62
Sample Complexity $x$	$x < 0.035$	$x < 0.11$	$x < 0.11$	$x < 0.26$

calculated respect to the based model of  $Trap_5$  function. The number of evaluation defer between test functions, and between population size (see Tables 2 and 3), and grows proportional to the difficulty of the test functions, as expected. The size of the cliques grows proportional to the structure of the test problem (see Tables 2 and 3). The similarity index indicates how the algorithm learned the structure of each test problem (see Tables 2 and 3). Observe that the test problem  $F_{c_2}$ , is not so well learned by the model (the similarity index =0.58). Remember that the algorithm assumes that the interaction structure of the test problem is at least a tree.

## 6 Discussion and Conclusions

The results of the experiments confirm that the first three problems are ease for the algorithm as was expected. The algorithm performance in the case of the last test problem (overlapping Traps) is very satisfactory too, meaning it that the linkage learning the using the predicted graphical Markov model and generating population samples by Cliques Gibbs Sampler, is an adequate structure for this more challenging complex problem for the test problem. This results confirm the Holland affirmation [8] chapter 1, about the complexity of these types of problems. Some part of the complexity comes from the effects of interaction between variables. Some comes from the effects of interactions within variables subsets, and some comes from the non additivity of the effects -a mentioned phenomenon known as epistasis.

As results of the experiment is to emphasize that the difficulty of the optimization, as assessed by the number of evaluation, is proportional to the sizes of the cliques of the learned models, that in time, is proportional to the structure of the test problem.

**Acknowledgements.** We would like to acknowledge support for this project (PIINF12-8) from the Autonomous University of Aguascalientes, Aguascalientes, Mexico.

## References

1. Akaike, H.: A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control* 19(6), 716–723 (1974)
2. Berge, C.: *Graphs and Hypergraph*. North-Holland, Amsterdam (1976)
3. Besag, J.E.: Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Royal Statist. Soc. Series B.* 36, 192–326 (1974)
4. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* IT-14(3), 462–467 (1968)
5. De la Maza, M., Tidor, B.: An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. In: *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 124–131. Morgan Kaufmann Publishers Inc., San Francisco (1993)
6. Diaz, E., Ponce-de-Leon, E., Larrañaga, P., Bielza, C.: Probabilistic Graphical Markov Model Learning: An Adaptive Strategy. In: Aguirre, A.H., Borja, R.M., Garcíá, C.A.R. (eds.) *MICAI 2009*. LNCS, vol. 5845, pp. 225–236. Springer, Heidelberg (2009)

7. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the bayesian distribution of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741 (1984)
8. Holland, J.H.: *Adaptation in natural and artificial system*, pp. 11–12. Univ. of Michigan Press, Ann. Arbor (1975/1992)
9. Kindermann, R., Snell, J.L.: *Markov random fields and their applications*. American Mathematical Society, Contemporary Mathematics, Providence, RI (1980)
10. Kruskal, J.B.: On the Shortest Spanning Tree of a Graph and the Traveling Salesman Problem. *Proceeding American Mathematical Society* 7, 48–50 (1956)
11. Lauritzen, S.L.: *Graphical models*. Oxford University Press, USA (1996)
12. Mühlenbein, H.: The equation for the response to selection and its use for prediction. *Evolutionary Computation* 5(3), 303–346 (1997)
13. Mühlenbein, H., Mahnig, T., Ochoa Rodriguez, A.: Schemata, Distributions and Graphical Models in Evolutionary Optimization. *Journal of Heuristic* 5(2), 215–247 (1999)
14. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions I. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) *PPSN 1996*. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
15. Munetomo, M., Goldberg, D.E.: Linkage Identification by Non - monotonicity Detection for Overlapping Functions. *Evolutionary Computation* 7(4), 377–398 (1999)
16. Pelikan, M.: *Bayesian Optimization Algorithm: From Single Level to Hierarchy*. University Illinois at Urbana Champaign, PHD Thesis. Also IlliGAL Report No. 2002023 (2002)
17. Pelikan, M., Mühlenbein, H.: The bivariate marginal distribution algorithm. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) *Advances in Soft Computing - Engineering Design and Manufacturing*, pp. 521–535. Springer, London (1999)
18. Ponce-de-Leon-Senti, E., Diaz, E.: Adaptive Evolutionary Algorithm based on a Cliques Gibbs Sampling over Graphical Markov Model Structure. In: Shakya, S., Santana, R. (eds.) *Markov Networks in Evolutionary Computation. Adaptation, Learning, and Optimization*, vol. 14, pp. 109–123. Springer, Heidelberg (2012)
19. Santana, R., Mühlenbein, H.: Blocked Stochastic Sampling versus Estimation of Distribution Algorithms. In: *Proceedings of the 2002 Congress on the Evolutionary Computation, CEC 2002*, vol. 2, pp. 1390–1395. IEEE Press (2002)
20. Winter, P.C., Hickey, G.I., Fletcher, H.L.: *Instant Notes in Genetics*, 2nd edn. Springer, New York (2002)

# A Comparison Study of PSO Neighborhoods

Angel Eduardo Muñoz Zavala

**Abstract.** A comparative study is performed to reveal the convergence characteristics and the robustness of three local neighborhoods in the particle swarm optimization algorithm (PSO): ring, Von Neumann and singly-linked ring. In the PSO algorithm, a neighborhood enables different communication paths among its members, and therefore, the way the swarm searches the landscape. Since the neighborhood structure changes the flying pattern of the swarm, convergence and diversity differ from structure to structure. A set of controlled experiments is developed to observe the transmission behavior (convergency) of every structure. The comparison results illustrate similarities and differences in the three topologies. A brief discussion is provided to further reveal the reasons which may account for the difference of the three neighborhoods.

## 1 Introduction

The particle swarm optimization (PSO) algorithm is a population-based optimization technique inspired by the motion of a bird flock [7]. Such groups are social organizations whose overall behavior relies on some sort of communication between members. Any member of the flock is called a “particle”. Most models for flock’s motion are based on the interaction between the particles, and the motion of the particle as an independent entity. In PSO, the particles fly over a real valued  $n$ -dimensional search space, where each particle has three attributes: position  $x$ , velocity  $v$ , and best position visited after the first fly  $P_{Best}$ . The best of all  $P_{Best}$  values is called global best  $G_{Best}$ . Its position is communicated to all flock members such that, at the time of the next fly, all the particles are aware of the best position visited. By “flying” a particle we mean to apply the effect of the local and global attractors

---

Angel Eduardo Muñoz Zavala

Universidad Autónoma de Aguascalientes, Ave. Universidad 940,

Colonia Ciudad Universitaria, CP 20131, Aguascalientes, AGS, México

e-mail: [aemz@correo.uaa.mx](mailto:aemz@correo.uaa.mx)

to the current motion vector. As a result, every particle gets a new position. The flock must keep flying and looking for better positions even when the current one seems good. Thus, finding the next best position is the main task of the flock for which exploration and therefore population diversity is crucial.

In PSO, the source of diversity, called *variation*, comes from two sources. One is the difference between the particle's current position and the global best, and the other is the difference between the particle's current position and its best historical value.

$$v_{t+1} = w * v_t + c_1 * U(0, 1) * (x_t - P_{Best}) + c_2 * U(0, 1) * (x_t - L_{Best}) \quad (1)$$

$$x_{t+1} = x_t + v_{t+1} \quad (2)$$

The equation above reflects the three main features of the PSO paradigm: distributed control, collective behavior, and local interaction with the environment [3]. The first term is the previous velocity (inertia path), the second term is called the cognitive component (particle's memory), and the last term is called the social component (neighborhood's influence).  $w$  is the inertia weight, and  $c_1$  and  $c_2$  are called acceleration coefficients.

The whole flock moves following the leader, but the leadership can be passed from member to member. At every PSO iteration the flock is inspected to find the best member. Whenever a member is found to improve the function value of the current leader, that member takes the leadership.

A leader can be global to all the flock, or local to a flock's neighborhood. In the latter case there are as many local leaders as neighborhoods. Having more than one leader in the flock translates into more attractors, possibly scattered over the search space. Therefore, the use of neighborhoods is a natural approach to fight premature convergence in the PSO algorithm [13].

For updating the particle position, the equation for local best PSO is similar to that of the global best PSO. One would simply substitute  $G_{Best}$  by  $L_{Best}$  in Equation 2.

$$v_{t+1} = w * v_t + c_1 * U(0, 1) * (x_t - P_{Best}) + c_2 * U(0, 1) * (x_t - L_{Best}) \quad (3)$$

In the PSO algorithm, a neighborhood structure enables different communication paths among its members, and therefore, the way the swarm searches the landscape. Since the neighborhood topology changes the flying pattern of the swarm, convergence and diversity differ from structure to structure.

This work studies three effective PSO neighborhoods: ring [8], Von Neumann [9] and singly-linked ring [16]. In Section 2 a brief analysis of these structures is

presented. The developed experiments are described in Section 3. The results of the performed comparison is discussed in Section 4. Finally, in Section 5 the conclusions are presented.

## 2 Neighborhood Topologies

There are several neighborhood structures for PSO. Their importance have been addressed by several researchers [7], [12], [14], [15]. Flock neighborhoods have a structure which define the way the information flows among members. Virtually, the information of the best particle, or leader, is concentrated and then distributed among its members.

The organization of the flock affects search and convergence capacity. The first particle swarm version used a kind of fully connected structure that became known as global PSO (Gbest) [7]. The fully connected structure has reported the fastest convergence speed [3, 8]. In a fully connected structure all the particles are neighbors of each other. The communication between particles is expeditious; thereby, the flock moves quickly toward the best solution found. Nevertheless, on non-smooth functions, the population will fail to explore outside of locally optimal regions. Namely, if the global optimum is not close to the best particle, it may be impossible to the swarm to explore other areas; this means that the swarm can be trapped in local optima.

The local PSO (Lbest) was proposed as a way to deal with more difficult problems. It offered the advantage that subpopulations could search diverse regions of the problem space [10]. In the local PSO, only a specific number of particles  $N_k$  (neighborhood) can influence the motion of a given particle  $k$ . Every particle is initialized with a permanent label which is independent of its geographic location in space. The swarm will converge slower but can locate the global optimum with a greater chance.

Various types of local PSO topologies are investigated and presented in literature [6]. Kennedy and Mendes compared traditional gbest topology to some lbest topologies like von Neumann, star, ring and pyramid [9]. They also have suggested a new methodology for involving neighborhood in PSO, called Fully-Informed Particle Swarm, which uses some portion of each neighbor's findings instead of the best neighbor and the best experience of the particle [10]. They have indicated that the individual's experience tends to be overwhelmed by social influence.

In [19], Safavieh et al. apply Voronoi diagram, which supports geometric dynamic neighborhood, to choose neighbors in PSO algorithm. The running time of the algorithm depends on the dimension of search space. By increasing the dimension the running time of the computing the Voronoi neighbors will increase.

Researchers who have suggested methods that use neighborhood, discussed that by this kind of neighborhood, a society is constructed between particles [19].

Once a particle finds a good result, it reports its location to its friends, and by some iterations all of the particles know something about good locations and try to move to them. The neighborhoods topologies study in this paper are: ring, Von Neumann and singly-linked ring.

### 2.1 Ring Topology

Flock members organized in a ring structure communicate with  $N_k$  immediate neighbors,  $N_k/2$  on each side (usually  $N_k = 2$ ). Finding the local best  $L_{Best}$  neighbor of particle  $k$  is done by inspecting the particles in the neighborhood:  $k + 1, k + 2, \dots, k + n/2$  and  $k - 1, k - 2, \dots, k - n/2$ .

In a ring structure, the information is slowly distributed among the flock members. This behavior does not contribute necessarily to improve the performance because it may be very inefficient during the refining phase of the solutions. However, using the ring topology will slow down the convergence rate because the best solution found has to propagate through several neighborhoods before affecting all particles in the swarm [4]. This slow propagation will enable the particles to explore more areas in the search space and thus decreases the chance of premature convergence [8].

The ring topology is used by most PSO implementations. In its simplest version, every particle  $k$  has two neighbors, particles  $k - 1$  and  $k + 1$ . Likewise, particles  $k - 1$  and  $k + 1$  have particle  $k$  as a neighbor. Therefore, there is a mutual attraction between consecutive particles because they are shared by two neighborhoods. This can be represented as a doubly-linked list [16], as shown in Figure 1

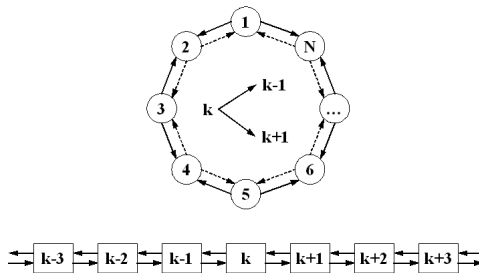


Fig. 1 Ring topology of neighborhood  $N_k = 2$

### 2.2 Von Neumann Topology

The Von Neumann topology, so-named after its use in cellular automata pioneered by John Von Neumann, was proposed for the PSO algorithm by Kennedy

and Mendes [9]. In a Von Neumann neighborhood, particles are connected using a rectangular matrix and every particle is connected to the individuals above, below, and to each side of it, wrapping the edges [10].

Kennedy and Mendes analyzed the effects of various neighborhoods structures on the PSO algorithm [9]. They recommended the Von Neumann structure because performed more consistently in their experiments than other neighborhoods tested [9].

In a Von Neumann topology every particle  $k$  has four neighbors, particles  $k - 1$  and  $k + 1$  at the sides, and particles  $k - \delta$  and  $k + \delta$  at the top and bottom, where  $\delta$  is a distance determined by the flock size. Likewise, particles  $k - 1, k + 1, k - \delta$  and  $k + \delta$  have particle  $k$  as a neighbor. Therefore, the Von Neumann topology possesses a doubly-linked list, like ring topology [16]. Figure 2.2 illustrates the Von Neumann neighborhood topology.

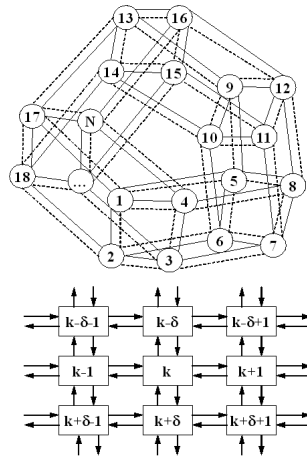


Fig. 2 Von Neumann topology

### 2.3 Singly-Linked Ring Topology

The singly-linked ring topology, introduced by Muñoz-Zavala et al. [16], organizes the flock in a modified ring fashion. This topology avoids the double list, that there are in the ring and Von Neumann topologies, applying an asymmetric connection. Algorithm 1 shows the procedure to find the neighbors for particle  $k$  in a singly-linked ring topology; where  $N_k$  is the neighborhood, and  $P_{(k+m)}$  is the particle located  $m$  positions beyond particle  $k$ .

As it is shown in Figure 3, in the singly-linked ring topology, every particle  $k$  has particles  $k - 2$  and  $k + 1$  as neighbors (not  $k - 1$  and  $k + 1$  as in the ring topology). In turn, particle  $k + 1$  has particles  $k - 1$  and  $k + 2$  as neighbors, and  $k - 1$  has



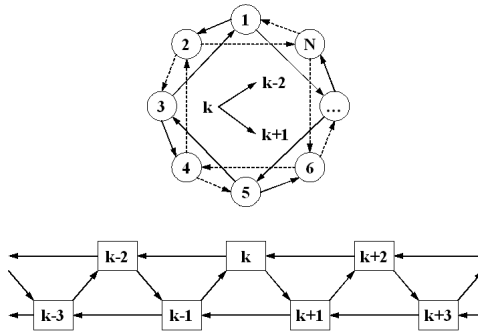
---

**Algorithm 1** Singly-linked ring neighborhood

---

- 1:  $N_k = \emptyset$
  - 2:  $Step = 1$
  - 3:  $Switch = 1$
  - 4: **repeat**
  - 5:    $N_k = N_k \cup P_{(k+Switch*Step)}$
  - 6:    $Step = Step + 1$
  - 7:    $Switch = -1 * Switch$
  - 8: **until**  $N_k = N$
- 

particles  $k - 3$  and  $k$  as neighbors. Then,  $k$  attracts  $k - 1$ , but  $k - 1$  only attracts  $k$  through particle  $k + 1$ . Therefore, the particle in between cancels the mutual attraction. Additionally, the information of the leader is transmitted to the particles at a lower speed [16].



**Fig. 3** Singly-linked ring topology of neighborhood  $N_k = 2$

The finding reported by Muñoz-Zavala et al. [16] is that by reducing the transfer information speed, the singly-linked ring keeps the exploration of the search space. The singly-linked ring structure allows neighborhoods of size  $N_k = flock/2 - 1$  without mutual attraction.

### 3 Experiments

A controlled test set was developed to show the convergence of each algorithm. The objective is to perform a fair comparison between the 3 structures using the same PSO model with the same parameters; thereby, a swarm size of 20 particles with a neighborhood size of  $N_k = 4$  particles is used for every topology. The neighbors of each particle  $k = \{1, \dots, 20\}$  are listed in Table 1 for every topology: ring (R), Von Neumann (VN) and singly-linked ring (SLR).

**Table 1** Neighbors of size  $N_k = 4$  for every topology

Particle	R	VN	SLR
1	2,3,19,20	2,4,5,17	2,4,17,19
2	1,3,4,20	1,3,6,18	3,5,18,20
3	1,2,4,5	2,4,7,19	1,4,6,19
4	2,3,5,6	1,3,8,20	2,5,7,20
5	3,4,6,7	1,6,8,9	1,3,6,8
6	4,5,7,8	2,5,7,10	2,4,7,9
7	5,6,8,9	3,6,8,11	3,5,8,10
8	6,7,9,10	4,5,7,12	4,6,9,11
9	7,8,10,11	5,10,12,13	5,7,10,12
10	8,9,11,12	6,9,11,14	6,8,11,13
11	9,10,12,13	7,10,12,15	7,9,12,14
12	10,11,13,14	8,9,11,16	8,10,13,15
13	11,12,14,15	9,14,16,17	9,11,14,16
14	12,13,15,16	10,13,15,18	10,12,15,17
15	13,14,16,17	11,14,16,19	11,13,16,18
16	14,15,17,18	12,13,15,20	12,14,17,19
17	15,16,18,19	1,13,18,20	13,15,18,20
18	16,17,19,20	2,14,17,19	1,14,16,19
19	1,17,18,20	3,15,18,20	2,15,17,20
20	1,2,18,19	4,16,17,19	1,3,16,18

In the field of evolutionary computation, it is common to compare different algorithms using a large test set. However, the effectiveness of an algorithm against another algorithm cannot be measured by the number of problems that it solves better [17]. The “no free lunch” theorem [22] shows that, if we compare two search algorithms with all possible functions, the performance of any two algorithms will be, on average, the same.

That is the reason why, when an algorithm is evaluated, we must look for the kind of problems where its performance is good, in order to characterize the type of problems for which the algorithm is suitable. In the experiments, we used three test functions well-known in the state-of-the-art and also used in [11]: Sphere, Rastrigin and Schwefel.

- Sphere function, proposed by De Jong [5] is a unimodal function and so very easy to find the minimum. It is a continuous and strongly convex function.

$$f(x) = \sum_{j=1}^2 x_j^2$$

$$x_j \in [-100, 100], \quad f^*(0, 0) = 0$$

- The Rastrigin function was constructed from Sphere adding a modulator term  $\alpha \cdot \cos(2\pi x_i)$ . It was proposed by Rastrigin [18] and is considered as difficult

for most optimization methods [17]. Rastrigin function has the property to have many local minima whose value increases with the distance to the global minimum.

$$f(x) = \sum_{j=1}^2 (x_j^2 - 10\cos(2\pi x_j) + 10)$$

$$x_j \in [-5.12, 5.12], \quad f^*(0, 0) = 0$$

- The Schwefel function is the hardest among three functions [20]. The surface of Schwefel function is composed of a great number of peaks and valleys. The function has a second best minimum far from the global minimum where many search algorithms are trapped [17]. Besides, the global minimum is near the bounds of the domain.

$$f(x) = \sum_{j=1}^2 (x_j \sin(\sqrt{|x_j|}) + 418.9829)$$

$$x_j \in [-500, 500], \quad f^*(420.96874, 420.96874) = 0$$

### 3.1 Test I

In the first test, we use the Sphere function with the initial particle positions presented in Table 2.

**Table 2** Initial positions for test I

Particle	$x_1$	$x_2$	Particle	$x_1$	$x_2$
1	0	0	11	50	50
2	5	5	12	55	55
3	10	10	13	60	60
4	15	15	14	65	65
5	20	20	15	70	70
6	25	25	16	75	75
7	30	30	17	80	80
8	35	35	18	85	85
9	40	40	19	90	90
10	45	45	20	95	95

In test I, particles are linearly arranged at different contour levels. The global optimum is assigned to particle  $k = 1$ . The aim is to illustrate the topology effect in the swarm motion.

### 3.2 Test II

The second test applies the initial particle positions given in Table 3 in the Rastrigin function.

**Table 3** Initial positions for test II

Particle	$x_1$	$x_2$	Particle	$x_1$	$x_2$
1	1	1	11	5	-5
2	-1	-1	12	-5	5
3	2	2	13	4	-4
4	-2	-2	14	-4	4
5	3	3	15	3	-3
6	-3	-3	16	-3	3
7	4	4	17	2	-2
8	-4	-4	18	-2	2
9	5	5	19	1	-1
10	-5	-5	20	-1	1

In test II, particles are grouped by contours levels (4 particles in each contour level), whose function value  $f(x_1, x_2)$  are approximately in local minima. The test try to show the ability of each topology to handle symmetric multimodal functions.

### 3.3 Test III

This test uses the Schwefel function with the initial particle positions shown in Table 4.

In test III, particles are grouped by neighborhood according Table 1. The test try to show the ability of each topology to handle asymmetric multimodal functions. The aim is to know the robustness of the studied neighborhood structures to find a global optimum away from the center.

## 4 Comparative Study

The aim is to perform an unbiased study. Thereby, we use the same PSO parameter in the next three local PSO approaches: PSO-R (ring topology), PSO-VN (Von Neumann topology) and PSO-SLR (singly-linked ring topology).

Van den Bergh analyzed PSO's convergence [21] and provided a model to obtain convergent trajectories  $0.5(c_1 + c_2) < w$ . In his work, Van den Bergh used the standard parameter values proposed by Clerc and Kennedy [2], and also applied by Bratton and Kennedy [1]:  $w = 0.7298$ ,  $c_1 = 1.49618$ ,  $c_2 = 1.49618$ .

**Table 4** Initial positions for test III

	PSO-R	PSO-VN	PSO-SLR
1	(420, -305)	(420, -305)	(420, -305)
2	(420, -305)	(420, -305)	(420, -305)
3	(420, -305)	(-305, -305)	(-305, -305)
4	(-305, -305)	(420, -305)	(420, -305)
5	(-305, -305)	(420, -305)	(-305, -305)
6	(-305, -305)	(-305, 420)	(-305, 420)
7	(-305, -305)	(-305, -305)	(-305, -305)
8	(-305, 420)	(-305, -305)	(-305, 420)
9	(-305, 420)	(-305, 420)	(-305, -305)
10	(-305, 420)	(-305, 420)	(-305, 420)
11	(-305, 420)	(-305, 420)	(-305, 420)
12	(-305, 420)	(-305, -305)	(-305, -305)
13	(-305, -305)	(-305, -305)	(-305, 420)
14	(-305, -305)	(-305, 420)	(-305, -305)
15	(-305, -305)	(-305, -305)	(-305, -305)
16	(-305, -305)	(-305, -305)	(-305, -305)
17	(-305, -305)	(420, -305)	(420, -305)
18	(-305, -305)	(-305, -305)	(-305, -305)
19	(420, -305)	(-305, -305)	(420, -305)
20	(420, -305)	(-305, -305)	(-305, -305)

#### 4.1 Comparative Test I

For each PSO algorithm (PSO-SLR, PSO-Ring and PSO-VN) 1000 runs were developed. Each algorithm performs the function evaluations required to reach a  $P_{Best} \leq 1 \times 10^{-16}$  for every particle  $k$ . The results are shown in Table 5.

**Table 5** Convergence results for test I

	PSO-R	PSO-VN	PSO-SLR
Mean	7027.86	7008.68	6997.38
Median	6980	6940	6940
Min	5500	5540	5340
Max	9240	10360	10020
Std. Desv.	601.55	588.34	604.11

The fitness evaluations performed by the 3 PSO algorithms to reach a  $P_{Best} \leq 1E - 16$  in the whole swarm are similar. In fact, there are insignificant differences for the mean, median and standard deviation values, between the 3 topologies. Table 6 presents the average function evaluations required for every particle to reach a  $P_{Best} \leq 1E - 16$ .

Table 6 shows that there are significant difference in the convergence results. For instance, particles  $k = 2$  and  $k = 3$  presents a difference in the average convergence results between the singly-linked ring topology and the other two topologies. In the

**Table 6** Average convergence results of each particle for test I

Particle	PSO-R	PSO-VN	PSO-SLR
1	0	0	0
2	203.17	201.387	223.676
3	211.784	217.691	212.698
4	217.236	215.054	226.659
5	223.457	218.267	220.337
6	226.445	227.273	234.319
7	231.381	233.364	233.243
8	236.835	237.91	242.043
9	241.988	249.668	245.614
10	246.318	254.491	252.521
11	252.622	256.428	257.582
12	260.623	262.283	264.312
13	264.678	274.137	268.656
14	273.87	278.755	273.141
15	282.573	281.023	280.896
16	289.207	283.961	285.631
17	293.286	273.323	288.304
18	294.25	275.572	273.079
19	277.517	283.239	283.639
20	280.056	282.14	279.244

PSO-SLR, particle  $k = 3$  converges before particle  $k = 2$ , despite that particle  $k = 2$  is closer to the optimum than particle  $k = 3$  (see Table 2), because particle  $k = 1$  (global optimum) belongs to the neighborhood of particle  $k = 3$  (see Table 1). In the PSO-Ring, particle  $k = 1$  is neighbor of both particles,  $k = 2$  and  $k = 3$ ; thereby, there is not an advantage for any particle. Finally, particle  $k = 1$  belongs only to the neighborhood of particle  $k = 2$  in the PSO-VN algorithm.

### 4.2 Comparative Test II

The function evaluations required to reach a  $P_{Best} \leq 1E - 16$ , for the whole swarm, were performed in each PSO algorithm described above. The results are shown in Table 7.

**Table 7** Convergence results for test II

	PSO-R	PSO-VN	PSO-SLR
Mean	8097.22	7842.82	8127.3
Median	8040	7750	8070
Min	6240	6540	6780
Max	10900	11460	10600
Std. Desv.	676.59	623.70	617.86

The fitness evaluations performed by the PSO-SLR and PSO-R algorithms to reach a  $P_{Best} \leq 1E - 16$  in the whole swarm are similar. There is a small difference between the Von Neumann and the other topologies in the mean and median convergence results. Table 8 presents the average function evaluations required for every particle to reach a  $P_{Best} \leq 1E - 16$ .

**Table 8** Average convergence results of each particle for test II

Particle	PSO-R	PSO-VN	PSO-SLR
1	331.887	288.467	332.236
2	328.927	290.827	334.689
3	315.006	313.608	334.403
4	305.924	316.664	332.756
5	302.951	291.887	322.194
6	303.158	292.875	325.509
7	311.764	311.444	325.935
8	315.415	311.573	328.919
9	325.772	317.6	326.452
10	329.055	316.867	330.452
11	328.102	316.095	332.605
12	323.983	316.573	331.993
13	313.553	312.848	331.085
14	309.071	310.601	331.13
15	300.749	293.242	333.313
16	304.937	292.516	331.503
17	304.742	315.487	331.984
18	313.174	316.23	335.084
19	328.718	288.913	335.95
20	330.49	289.531	331.294

Table 8 shows that there are several differences in the convergence results between the 3 topologies. For instance, in the PSO-R the particles allocated in the contour level  $f(x_1, x_2) = 18$  (particles  $k = \{5, 6, 15, 16\}$ ) reach the global optimum faster than the particles allocated in contour levels nearby to the global optimum (see Table 3). Similarly, in the PSO-VN, these particles reach the global optimum straightway the particles in the contour level  $f(x_1, x_2) = 2$  (particles  $k = \{1, 2, 19, 20\}$ ), outperform the average convergence results of particles  $k = \{3, 4, 17, 18\}$  allocated in the contour level  $f(x_1, x_2) = 8$ . On the other hand, the PSO-SLR convergence results are very similar for all the particles, we may say that the swarm “flies together”. Besides, the singly-linked ring shows a slower motion than the other two topologies.

### 4.3 Comparative Test III

Each PSO algorithm performs the function evaluations required to reach a  $P_{Best} \leq 1E - 04$  for every particle  $k$ . If the algorithm is trapped in local minima, the run is finished at 20,000 function evaluations and reported as unsuccessful run. The results are shown in Table 9.

**Table 9** Convergence results for test III

	PSO-R	PSO-VN	PSO-SLR
Mean	3966.46	3962.40	3922.04
Median	3800	3740	3700
Min	2580	2700	2760
Max	18740	19580	14920
Std. Desv.	1192.91	1266.26	996.14
Unsuccessful Run	3.1%	3.4%	0.9%

The fitness evaluations performed by the PSO-R and PSO-VN algorithms to reach a  $P_{Best} \leq 1E - 04$  in the whole swarm are similar. In fact, there are insignificant differences for the mean, median and standard deviation values, between the

**Table 10** Average convergence results of each particle for test III

Particle	PSO-R	PSO-VN	PSO-SLR
1	141.1987891	139.8146998	139.623323
2	139.9566095	139.4658385	139.3065015
3	141.2926337	139.373706	138.7997936
4	140.8193744	138.8022774	139.2693498
5	140.5005045	141.245614	140.7997936
6	139.7719475	138.7225673	139.8167702
7	141.2431887	139.8799172	139.1671827
8	141.1210898	138.6335404	139.4427245
9	140.7830474	141.5093168	140.2786378
10	141.6801211	138.4633643	139.5614035
11	139.2179617	140.0786749	139.0890269
12	139.6528759	140.7401656	139.380805
13	140.938446	141.1362229	139.7089783
14	140.1029263	138.1697723	140.0310559
15	138.7860747	138.073499	140.2641899
16	138.5095863	139.9285714	139.5675955
17	137.7134208	140.4761905	140.4499484
18	137.864783	139.2142857	140.2373581
19	138.7093845	138.8395445	139.6955624
20	139.6801211	139.1242236	139.1795666



2 topologies. Nevertheless, the PSO-SLR results outperform significantly the standard deviation and successful runs values obtained by the other two topologies. Table 10 presents the average function evaluations required for every particle to reach a  $P_{Best} \leq 1E - 04$ .

Table 10 shows that there is not a significant difference in the convergence results for all the swarm between the topologies tested. These results are consistent with the mean and median convergence results show in Table 9. Nevertheless, we can not find a relation between these results and the unsuccessful run rates show in Table 9. There is a marked difference in the PSO-SLR successful rate 99.1% to reach the global optimum against the successful rates 96.9% and 96.4%, obtained by the PSO-R and PSO-VN, respectively.

The results obtained in test I and test II can help us to understand the effect of the topology structure in the communication of the swarm, which is illustrated in the unsuccessful rate results obtained with a singly-linked ring topology. Muñoz-Zavala et al. [16] conclude that the singly-linked ring topology outperforms the ring and Von Neumann topologies due to the double list that exist between the particles in these topologies; which are eliminated in their approach.

## 5 Conclusions

In the PSO algorithm the communication between particles is essential. The information is transmitted according to an interaction structure (neighborhood), which can be global or local. The neighborhood affects the transmission speed and influences the PSO convergence.

This papers proposed a study comparison between three topologies (neighborhood structures) which are the *state-of-the-art*: ring, Von Neumann and singly-linked ring. Three controled tests were applied to illustrate the global and individual convergence of the PSO algorithm applying the 3 topologies. The convergence results show that the global convergence is similar for the 3 topologies. On the other hand, the individual (particle) convergences are different in the 3 neighborhood approaches.

Although the singly-linked ring outperforms the ring and Von Neumann topologies in test III, the tests are not conclusive and more experiments are being performed. A future work is to apply the topologies studied in this paper in another kind of functions. An analysis of indirect neighbors may provide additional information about swarm motion mechanism.

## References

1. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 120–127. IEEE (2007)
2. Clerc, M., Kennedy, J.: The particle swarm: explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)

3. Eberhart, R., Dobbins, R., Simpson, P.: Computational Intelligence PC Tools. Academic Press Professional (1996)
4. Hamdan, S.A.: Hybrid particle swarm optimiser using multi-neighborhood topologies. *INFOCOMP Journal of Computer Science* 7, 36–44 (2008)
5. De Jong, K.D.: An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, USA (1975)
6. Kennedy, J.: Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1931–1938. IEEE (1999)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948. IEEE (1995)
8. Kennedy, J., Eberhart, R.: *The Particle Swarm: Social Adaptation in Information-Processing Systems*. McGraw-Hill, London (1999)
9. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: *Proceedings of the 2002 Congress on Evolutionary Computation*, pp. 1671–1676. IEEE (2002)
10. Kennedy, J., Mendes, R.: Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms. *IEEE Transaction on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 36(4), 515–519 (2006)
11. Kim, Y.H., Lee, K.H., Yoon, Y.: Visualizing the search process of particle swarm optimization. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 49–55. ACM (2009)
12. Mendes, R.: Population topologies and their influence in particle swarm performance. PhD thesis, Escola de Engenharia, Universidade do Minho (2004)
13. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation* 8(3), 204–210 (2004)
14. Mendes, R., Neves, J.: What Makes a Successful Society? Experiments with Population Topologies in Particle Swarms. In: Bazzan, A.L.C., Labidi, S. (eds.) *SBIA 2004*. LNCS (LNAI), vol. 3171, pp. 346–355. Springer, Heidelberg (2004)
15. Mohai, A., Mendes, R., Ward, C., Posthoff, C.: Neighborhood re-structuring in particle swarm optimization. In: *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, pp. 776–785. Springer (2005)
16. Muñoz-Zavala, A.E., Hernández-Aguirre, A., Villa-Diharce, E.R.: The singly-linked ring topology for the particle swarm optimization algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 65–72. ACM (2009)
17. Ortiz-Boyer, D., Hervás-Martínez, C., García, N.: Cix12: A crossover operator for evolutionary algorithms based on population features. *Journal of Artificial Intelligence Research* 24, 1–48 (2005)
18. Rastrigin, L.A.: Extremal control systems. *Cybernetics Series*. In: *Theoretical Foundations of Engineering*, Nauka, Russian (1974)
19. Safavieh, E., Gheibi, A., Abolghasemi, M., Mohades, A.: Particle swarm optimization with Voronoi neighborhood. In: *Proceedings of the International CSI Computer Conference 2009*, pp. 397–402. IEEE (2009)
20. Schwefel, H.P.: *Numerical optimization of computer models*. John Wiley and Sons, New York (1981)
21. Van den Bergh, F.: An analysis of particle swarm optimizers. PhD thesis, University of Pretoria, South Africa (2002)
22. Wolpert, D.H., Macready, W.G.: No free-lunch theorems for search. Technical report, Santa Fe Institute (1995)

# *hyp*DE: A Hyper-Heuristic Based on Differential Evolution for Solving Constrained Optimization Problems

José Carlos Villela Tinoco and Carlos A. Coello Coello

**Abstract.** In this paper, we present a hyper-heuristic, based on Differential Evolution, for solving constrained optimization problems. Differential Evolution has been found to be a very effective and efficient optimization algorithm for continuous search spaces, which motivated us to adopt it as our search engine for dealing with constrained optimization problems. In our proposed hyper-heuristic, we adopt twelve differential evolution models for our low-level heuristic. We also adopt four selection mechanisms for choosing the low-level heuristic. The proposed approach is validated using a well-known benchmark for constrained evolutionary optimization. Results are compared with respect to those obtained by a state-of-the-art constrained differential evolution algorithm (CDE) and another hyper-heuristic that adopts a random descent selection mechanism. Our results indicate that our proposed approach is a viable alternative for dealing with constrained optimization problems.

## 1 Introduction

Heuristics have been a very effective tool for solving a wide variety of real-world problems having a very large and little known search space. In its origins, research on heuristics spent a great deal of efforts in designing generic heuristics that were meant to be superior to the others in all classes of problems. This effort radically switched after the publication of the *No Free Lunch Theorems* for search in the 1990s [16]. This work provided a mathematical proof of the impossibility to design a heuristic that can be better than all the others in all classes of problems. This led to a different type of research in which the focus switched to analyzing the strengths

---

José Carlos Villela Tinoco · Carlos A. Coello Coello  
CINVESTAV-IPN (Evolutionary Computation Group)  
Departamento de Computación, Av. IPN No. 2508  
Col. San Pedro Zacatenco, México, D.F. 07360, MÉXICO  
e-mail: [cocoello@cs.cinvestav.mx](mailto:cocoello@cs.cinvestav.mx)

and limitations of heuristics in particular classes of problems, aiming to identify the cases in which a certain type of heuristic may be better than others. These studies naturally led to the idea of combining the efforts of different heuristics into a single scheme. The motivation here would be to compensate the weaknesses of one heuristic with the strengths of another one in a certain type of problem. From the different research proposals in this direction, one of the most promising has been that of the *hyper-heuristics*, in which the idea is to design an approach that uses a control mechanism for selecting from among several possible low-level heuristics. The main motivation of hyper-heuristics is to release the user from the burden of selecting a particular heuristic for the problem at hand (something that tends to be cumbersome). Although hyper-heuristics have been mainly used in combinatorial optimization problems [5] we adopt this same framework here for constrained continuous optimization.

In this paper, we propose a new hyper-heuristic based on Differential Evolution (DE) variants which is aimed to solve constrained optimization problems in which the decision variables are real numbers. The main contribution of this work is a new selection mechanism designed to coordinate the different Differential Evolution models incorporated into the proposed hyper-heuristic.

The remainder of this paper is organized as follows. In Section 2, we provide a short description of the Differential Evolution algorithm. In Section 3, we briefly describe the origins of hyper-heuristics and their core idea. The previous related work is discussed in Section 4. Our proposed approach is provided in Section 5. Our experimental results are presented in Section 6. Finally, Section 7 presents our main conclusions and some possible paths for future research.

## 2 Differential Evolution

Differential Evolution (DE) was proposed in 1995 by Kenneth Price and Rainer Storn in 1995 as a new heuristic for optimization of nonlinear and non-differentiable functions [15]. In DE, the decision variables are assumed to be real numbers, and new solutions are generated by combining a parent with other individuals. The main DE algorithm can be defined based on the following concepts:

(i) **The population:**

$$P_{x,g} = (\mathbf{x}_{i,g}) \quad i = 0, 1, \dots, NP - 1, \quad g = 0, 1, \dots, G_{\max} \quad (1)$$

$$\mathbf{x}_{i,g} = [x_0, x_1, \dots, x_{D-1}]^T$$

where  $NP$  denotes the maximum number of vectors that make up the population,  $g$  is the generation counter,  $G_{\max}$  is the maximum number of generations and  $D$  is the number of decision variables of the problem.

(ii) **Mutation operator:**

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}), r_1 \neq r_2 \neq r_3 \neq i \quad (2)$$

where  $r_1, r_2$  and  $r_3 \in [1, NP]$  are randomly selected vectors.  $F > 0$  is a real value that controls the amplification of the difference vector and  $\mathbf{x}_{r_1,G}$  is the base vector.

(iii) **Crossover operator:**

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } U(0, 1) \leq Cr \text{ or } j = j_{\text{rand}} \\ x_{ji,G} & \text{otherwise} \end{cases} \quad (3)$$

where  $Cr \in [0, 1]$  is the crossover constant which has to be determined by the user and  $j_{\text{rand}}$  is a randomly chosen index  $\in 1, 2, \dots, D$ .

(iv) **Selection operator:**

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & \text{if } f(\mathbf{u}_{i,G+1}) < f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{otherwise} \end{cases} \quad (4)$$

To decide whether or not a solution should become a member of generation  $G + 1$ , the vector  $\mathbf{u}_{i,G+1}$  is compared to the vector  $\mathbf{x}_{i,G}$ ; if  $\mathbf{u}_{i,G+1}$  yields a smaller objective function value than  $\mathbf{x}_{i,G}$ , then  $\mathbf{x}_{i,G+1}$  takes the value  $\mathbf{u}_{i,G+1}$ ; otherwise, the old value is retained.

The main DE variants are named using the following notation:  $DE/x/y/z$ , where  $x$  represents the base vector to disturb,  $y$  is the number of pairs of vectors that are to be disturbed and  $z$  is the type of recombination to be adopted [11]. Algorithm 17 shows variant of DE called  $DE/rand/1/bin$ , which is the most popular in the specialized literature. *rand* indicates that the base vector to be disturbed is chosen at random and *bin* means that binomial recombination is adopted.

---

**Algorithm 1** Differential Evolution algorithm in its  $DE/rand/1/bin$  variant

---

```

G ← 0
Initialize  $P_{x,G}$ 
while Termination criterion not satisfied do
    for  $i \leftarrow \{0, \dots, NP - 1\}$  do
        Select  $r_1, r_2, r_3 \in \{0, \dots, NP - 1\}$  randomly, where  $r_1 \neq r_2 \neq r_3$ 
        Select  $j_{\text{rand}} \in \{0, \dots, D - 1\}$  randomly
        for  $j \leftarrow \{1, \dots, D - 1\}$  do
            if  $U[0, 1] < Cr$  or  $j = j_{\text{rand}}$  then
                 $u_{i,j} \leftarrow x_{r_3,j,G} + F(x_{r_1,j,G} - x_{r_2,j,G})$ 
            else
                 $u_{i,j} \leftarrow x_{i,j,G}$ 
            if  $f(\mathbf{u}_i) \leq f(\mathbf{x}_{i,G})$  then
                 $\mathbf{x}_{i,G+1} \leftarrow \mathbf{u}_i$ 
    G ← G + 1
    
```

---

### 3 Hyper-Heuristics

The use of heuristics for the solution of high complexity problems has become very popular in recent years, mainly because of their flexibility, efficacy and ease of use. However, this popularity has simultaneously fostered the development of a wide variety of heuristics. Such a diversity of methods makes it difficult to select one for a particular problem. Additionally, there are very few studies that attempt to identify the main advantages or disadvantages of a heuristic with respect to others, in a particular problem (or class of problems).

The term *hyper-heuristic* was originally introduced by Cowling et al. [5] to refer to approaches that operate at a higher level of abstraction than conventional heuristics. Additionally, a hyper-heuristic is capable of identifying which low-level heuristic needs to be used at a certain moment. In other words, hyper-heuristics operate in the **space of available heuristics** while heuristics work directly on the **space of solutions** of the problem [14]. Thus, a generic procedure for a hyper-heuristic is the following [2]:

- (i) **Step 1.** Start with a set  $H$  of heuristics each of which is applicable to a problem state and transforms it into a new problem state.
- (ii) **Step 2.** Let the initial problem state be  $S_0$ .
- (iii) **Step 3.** From the state  $S_i$  of the problem, find the most appropriate heuristic to transform the problem to the next state ( $S_{i+1}$ ).
- (iv) **Step 4.** If the problem has been solved, stop. Otherwise, go to Step 3.

The main aim of hyper-heuristics is to provide a general framework that can offer good quality solutions for a larger number of problems. This suggests that a hyper-heuristic that has been developed for a particular problem could be easily extended to other domains by simply replacing the set of low-level heuristics and the evaluation function [4]. There is, of course, a well-defined interface between the hyper-heuristic and its low-level heuristics in order to achieve this objective. This interface must consider the following aspects:

- (i) The interface should be standard, that is, only one interface is required to communicate the hyper-heuristic to the set of heuristics; otherwise, it will require a separate interface for each heuristic.
- (ii) The interface should facilitate its portability to other domains. When requiring to solve a new problem, the user only has to supply all the low-level heuristics and the corresponding evaluation function.

### 4 Previous Related Work

Hyper-heuristics have been mainly used in combinatorial optimization, and their use in continuous optimization problems is still rare (if we consider constrained problems, then their use is even more scarce). The only previous related work

that we found is the paper from Biazzi et al. [1] in which they proposed a distributed hyper-heuristic for solving unconstrained continuous optimization problems. These authors adopted an island model and distributed several low-level heuristics throughout the islands. The authors concluded that their proposed approach produced results that were more consistent than those obtained by any of the low-level heuristics adopted, when considered in an independent manner. Biazzi et al. [1] adopted six DE models, a particle swarm optimizer and a random sampling algorithm. Each island was assigned a population of size  $NP$  and implemented the following seven selection mechanisms:

- (i) **StatEq.**- assigns a heuristic to each island at the beginning of the run and does not change this assignment anymore.
- (ii) **DynEq.**- assigns a random heuristic to each island after each cycle, where one cycle within an island represents the generation of one new solution using a heuristic.
- (iii) **Tabu.**- corresponds to an adaptation to the hyper-heuristic proposed in [3] and it runs this algorithm on each of the islands that make up the hyper-heuristic.
- (iv) **SDigmo and DDigmo.**- assigns a probability of selecting each heuristic based on the performance of each of the algorithms so that, after that probability is computed, a heuristic can be assigned to each of the nodes (islands).
- (v) **Pruner.**- initially uses the entire collection of available algorithms, but as the search proceeds, it removes more and more algorithms from this set and does not consider them anymore.
- (vi) **Scanner.**- the algorithms are sorted based on the latest solutions they have found so far and defines a minimum number of consecutive executions for each heuristic in each island.

However, as mentioned above, the work of Biazzi et al. [1] does not include a constraint handling mechanism. This is precisely the issue that we address here: how to design a hyper-heuristic for constrained continuous optimization using DE variants as our low-level heuristics.

## 5 Our Proposed Approach

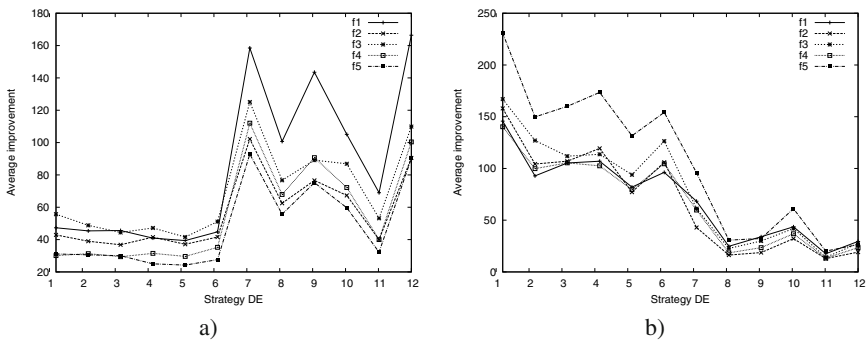
As indicated before, we propose here a new hyper-heuristic for constrained optimization (in continuous search spaces), based on the use of different DE variants. Table 1 shows each of the DE variants used as low-level heuristics for our proposed hyper-heuristic.

We adopted a selection mechanism that aims to incorporate some type of knowledge for choosing the low-level heuristics to be applied at any given time. For designing such a selection mechanism, it was necessary to identify some of the characteristics of each of the DE variants. This led us to implement a random descent mechanism [5] to identify the behaviors and characteristics of the different DE variants when applied to unconstrained optimization problems (a constraint-handling

**Table 1** Set of low-level heuristics used by our proposed hyper-heuristic

Models with <i>exp</i> recombination		Models with <i>bin</i> recombination	
$h_1$	DE/best/1/exp	$h_7$	DE/best/1/bin
$h_2$	DE/rand/1/exp	$h_8$	DE/rand/1/bin
$h_3$	DE/current-to-best/1/exp	$h_9$	DE/current-to-best/1/bin
$h_4$	DE/best/2/exp	$h_{10}$	DE/best/2/bin
$h_5$	DE/rand/2/exp	$h_{11}$	DE/rand/2/bin
$h_6$	DE/current-to-rand/1/exp	$h_{12}$	DE/current-to-rand/1/bin

mechanism is incorporated later on). Figure 1 shows the behavior of the DE variants adopted when using two different values of  $Cr$ . Here, we can observe that this parameter plays an important role on the performance of each DE variant. In fact, we found out as well that the type of recombination that performed better was related to the value of  $Cr$  that was adopted (i.e., for certain values of  $Cr$ , either the binary or the exponential recombination performed better). Additionally, we found a correlation between the type of recombination that was more effective and the number of decision variables of the problem. Then, for certain combinations of these elements ( $Cr$  value, recombination type, and dimensionality), we found out that a particular DE variant performed better. All of these results were found in our experimental study, but the details are omitted here due to space constraints.

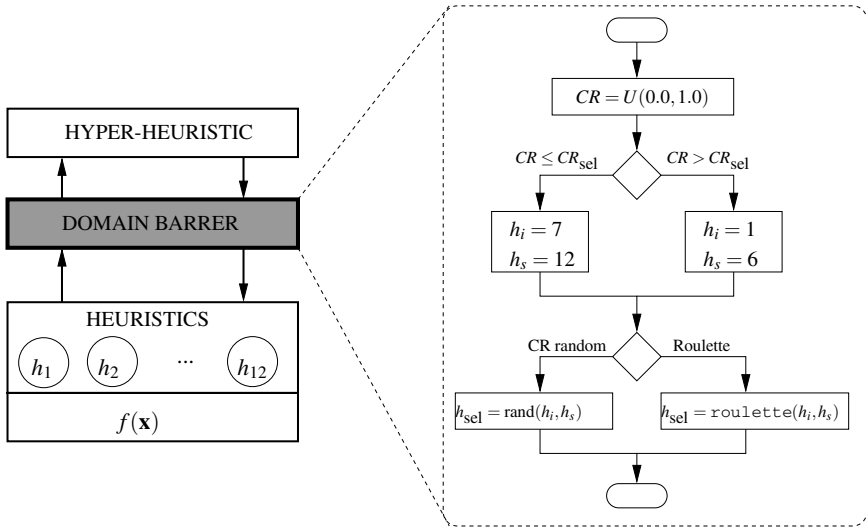
**Fig. 1** Total successful steps for each model when using: a)  $Cr = 0.2$  and b)  $Cr = 0.8$ 

Based on the experimental results previously indicated, our hyper-heuristic consists of two phases. The first phase is responsible for selecting the type of recombination to be adopted (either *exp* or *bin*). In order to do this, we randomly vary at each generation  $g$ , the parameter  $Cr$  within the range  $[0, 1]$  and, depending on the value of  $Cr$  that we adopt, and on the number of decision variables of the problem, we select the type of recombination to be used. Once we have decided what type of



recombination to use, the second phase consists in selecting the specific model to be applied for generating the population  $P_{x,g+1}$ .

For the second phase, we implemented two selection mechanisms. The first of them is called **Cr random**, and, as its name indicates, it consists of randomly choosing the DE variant to be used. The second mechanism uses a roulette wheel to choose the DE variant to be adopted. The diagram shown in Fig. 2 indicates the process for choosing the low-level heuristic to be applied.



**Fig. 2** Diagram that illustrates the selection mechanism of the proposed hyper-heuristic

When using a roulette wheel, a certain probability is assigned to each of the DE variants, based on their performance during the search process, so that the probability of selecting a certain DE variant is proportional to its performance. In order to mitigate the well-known bias problems of the roulette wheel selection mechanism (i.e., the worst individual may be selected several times [5]), we implemented three types of roulette wheel selection:

- **Original roulette (R1):** this is the original algorithm proposed in [6].
- **Roulette with random init (R2):** in this case, the initial position of the roulette wheel is randomly selected.
- **Roulette with permutation (R3):** in this case, we create a permutation of the positions of the roulette and then, we apply the original algorithm.

Algorithm 18 shows the pseudocode of the proposed selection mechanism. Here, *mechanism* refers to the selection mechanism and  $Cr_{sel}$  is a parameter indicating the probability of selecting a DE variant either with binomial or with exponential recombination. The parameter  $Cr_{sel}$  is calculated based on the equation (5), so that

the percentage of selection of DE variants with either exponential or binomial recombination is determined based on their performance on the problem being solved.

$$Cr_{sel} = \frac{\text{Total success of variants with binomial recombination}}{\text{Total success of all variants}}$$

$$Cr_{sel} = \frac{\sum_{i=7}^{12} success_i}{\sum_{i=1}^6 success_i + \sum_{i=7}^{12} success_i} \quad (5)$$

---

**Algorithm 2** Selection mechanism proposed
 

---

```

Input: mechanism
Output: selected heuristic
switch mechanism do
  case RANDOM
  | heuristicsel ← rand(h1, h12)
  case Cr_RANDOM
  | Cr ← U(0, 1)
  | if Cr > Crsel then
  | | /* DE variants with exponential
  | | recombination */
  | | heuristicsel ← rand(h1, h6)
  | else
  | | /* DE variants with binomial recombination
  | | */
  | | heuristicsel ← rand(h7, h12)
  case ROULETTE
  | Cr ← U(0, 1)
  | if Cr > Crsel then
  | | Select strategies with exponential recombination using a
  | | roulette-wheel
  | else
  | | Select strategies with binomial recombination using a
  | | roulette-wheel
  
```

---

Finally, Algorithm 19 shows the pseudocode of the proposed hyper-heuristic. The control parameters of the proposed algorithm are the following:

- $G_{\max}$ : maximum number of generations.
- $NP$ : number of individuals in the population.
- $Cr$ : DE's crossover constant.
- *mechanism*: type of selection mechanism adopted for the low-level heuristics incorporated within the hyper-heuristic.

It is important to note that at the beginning of the search process we do not have information about the performance of each low-level heuristic. Therefore, we require an initial *training stage*, which consists of the implementation of a maximum

number of generations in which we use the random descent selection mechanism proposed in [5] to initialize the expected values (EVs) for each of the DE variants adopted, according to the following equation:

$$EV_i = \frac{success_i}{\frac{1}{12} \sum_j^{12} success_j} \quad i = 1, 2, \dots, 12 \quad (6)$$

---

**Algorithm 3** Our proposed hyper-heuristic
 

---

**Input:**  $NP$ ,  $G_{\max}$ ,  $Cr$ , mechanism

$G \leftarrow 0$

Initialize  $P_{x,G}$

**if** mechanism is *Cr\_RANDOM* or *RULETTE* **then**

  /\* Training stage \*/

$G_{aux} \leftarrow 0$

**while**  $G_{aux} < G_{proof}$  **do**

    /\* Apply mechanism **random descent** \*/

    selection\_mechanism()

**repeat**

      apply\_heuristic( $heuristic_{sel}$ )

$G \leftarrow G + 1$

$G_{aux} \leftarrow G_{aux} + 1$

**until** not being able to improve the previous solution;

  Initialize the expected value of each low-level heuristic

**while** Termination criterion not satisfied **do**

  selection\_mechanism()

**if** mechanism *DESCENT* **then**

**repeat**

      apply\_heuristic( $heuristic_{sel}$ )

$G \leftarrow G + 1$

**until** not improve the previous solution;

**else**

    apply\_heuristic( $heuristic_{sel}$ )

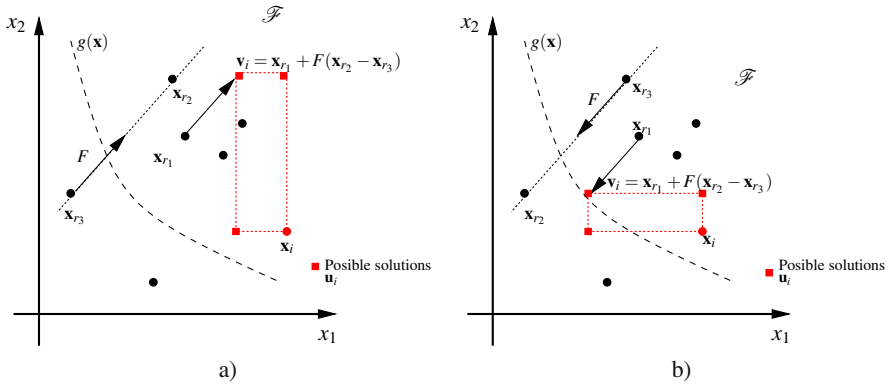
$G \leftarrow G + 1$

---

## 5.1 Handling Constraints

Differential Evolution was designed for solving unconstrained optimization problems. Thus, it is necessary to incorporate to it a constraint-handling scheme for dealing with constrained optimization problems. One of the most popular constraint-handling techniques used with evolutionary algorithms has been Stochastic Ranking (SR) [12]. SR adopts a rank selection mechanism that tries to balance the influence of considering either the objective function value or the degree of violation of the constraints (a parameter called  $P_f$  is adopted for this sake). SR has been successfully

incorporated into DE before. For example, [7] showed that when embedding SR into DE, this constraint-handling mechanism can provide information to the mutation operator about the most appropriate direction of movement. This produces, indeed, a speed up in the convergence of the algorithm. Figure 3 shows two hypothetical examples for the search directions to be considered: a) the movement of an infeasible point to the feasible region, and b) the movement of a feasible point to the infeasible region. This illustrates that it is possible to guide the search in such a way that we can generate new solutions either closer or farther away from the feasible region  $\mathcal{F}$  (corresponding to cases a) and b) in Figure 3 respectively).



**Fig. 3** Example of the two types search direction: a) towards the feasible region  $\mathcal{F}$  b) towards the infeasible region

Stochastic Ranking sorts the individuals in the population, and the rank of each individual (i.e., its position in the sorted list) is used to guide the search in a specific direction. In order to select the search direction, we adopt a probabilistic value that regulates the type of movements performed (either to move a solution into the feasible region or towards the infeasible region) and its use aims to explore the boundary between the feasible and the infeasible region. We experimentally found that this probabilistic value provided good results when set with the same value used by the  $P_f$  parameter of SR.

Based on the previous discussion, we modified the selection operator of DE using the following criteria:

- If the two solutions being compared are feasible, the one with the best objective function value is chosen.
- Otherwise, we choose the solution with the lowest degree of constraint violation.

Algorithm 20 shows the process carried out to generate, from the current population  $P_{x,G}$ , the new population  $P_{x,G+1}$ . The objective function is denoted by  $f$  and the degree of constraint violation is denoted by  $\phi$ . Here,  $\phi$  is given by:

$$\phi(\mathbf{x}) = \sum_{i=1}^m \max\{0, g_i(\mathbf{x})\}^2 + \sum_{j=1}^p |h_j(\mathbf{x})|^2 \quad (7)$$

where  $g_i(\mathbf{x})$  correspond to the inequality constraints and  $h_j(\mathbf{x})$  denote the equality constraints of the problem.

---

**Algorithm 4** Pseudocode of our proposed DE approach for constrained optimization

---

**Input:** strategy of DE to use

**Result:** generation  $G + 1$

**begin**

    Apply Stochastic Ranking to rank the population  $P_{x,G}$

**for**  $i \leftarrow \{1, \dots, NP\}$  **do**

        /\* Selecting vectors to disturb \*/

**if**  $U(0,1) < P_f$  **then**

            /\* To the infeasible region \*/

$r_1 \leftarrow \text{rand}(1, \alpha)$

$r_2 \leftarrow \text{rand}(\alpha + 1, NP)$

**else**

            /\* To the feasible region \*/

$r_1 \leftarrow \text{rand}(\alpha + 1, NP)$

$r_2 \leftarrow \text{rand}(1, \alpha)$

        Select vector  $r_3$  according to the DE strategy to be used:

        /\* Mutation \*/

$\mathbf{v}_{k,G} = \mathbf{x}_{k,G,r3} + F(\mathbf{x}_{k,G,r1} - \mathbf{x}_{k,G,r2})$

        /\* Recombination \*/

        Apply recombination operator according to the DE strategy adopted in order to obtain  $\mathbf{u}_i$

        /\* Selection \*/

**if**  $\phi(\mathbf{u}_i) = \phi(\mathbf{x}_{i,G}) = 0$  **then**

**if**  $f(\mathbf{u}_i) < f(\mathbf{x}_{i,G})$  **then**

$\mathbf{x}_{i,G+1} \leftarrow \mathbf{u}_i$

**else**

$\mathbf{x}_{i,G+1} \leftarrow \mathbf{x}_{i,G}$

**else**

**if**  $\phi(\mathbf{u}_i) < \phi(\mathbf{x}_{i,G})$  **then**

$\mathbf{x}_{i,G+1} \leftarrow \mathbf{u}_i$

**else**

$\mathbf{x}_{i,G+1} \leftarrow \mathbf{x}_{i,G}$

## 6 Results

In order to validate the performance of our proposed approach, we adopted several standard test functions from the specialized literature on evolutionary constrained optimization [12]. Our results were compared with respect to those generated by the approach called **CDE** (Constrained Differential Evolution) [10] which is representative of the state-of-the-art on DE-based constrained optimization. Additionally, we also compared results with respect to the hyper-heuristic with a random descent mechanism proposed in [5]. The hardware and software platform adopted for our experiments, as well as the parameters adopted are the following:

(i) PC configuration:

System: Linux Ubuntu 10.04

CPU: Intel Pentium Dual Core Inside T2080 (1.73 GHz)

RAM: 1024 MB

Programming Language: C (gcc 4.4.3 compiler)

(ii) Parameters:

Maximum number of generations:  $G_{\max} = 6000$

Population size:  $NP = 50$

$F : U(0.3, 0.9)$  each generation

$Cr : U(0, 1)$  each generation

$P_f : 0.50$  for *g06* and *g11*, and 0.45 for the remaining test problems

From the parameters adopted, it can be seen that all the approaches perform 300,000 objective function evaluations. Our experimental study comprised 100 independent runs per algorithm per problem. In Tables 2, 3 and 4 we show the results obtained by the CDE algorithm, the random descent mechanism and our proposed approach, using the three roulette-wheel selection mechanisms previously indicated (**R1** corresponds to the original roulette-wheel, **R2** corresponds to roulette-wheel with random initial position and **R3** corresponds to the roulette-wheel with permutation). From these results, it can be noticed that not all the approaches required the maximum number of generations to reach the best known result. That is the reason why in Table 5 we show the average number of generations in which the algorithm converges as well as the minimum number of generations that each algorithm required to find the best known solution for each test problem.

### 6.1 Analysis of Results

From the results presented in Tables 2, 3 and 3, we can see that the selection mechanism based on a roulette-wheel with random initial position (**R2**) obtained good results in most of the test problems. These results also indicate that the algorithm **CDE** obtained very poor results in the test problems *g03*, *g05*, *g10* and *g13*. We can also observe that the hyper-heuristic approaches were unable to solve *g11* in a proper way, which is not the case of the CDE algorithm.

**Table 2** Statistics with respect to the  $f(\mathbf{x})$  obtained by CDE and the random descent mechanism.  $\mu$  corresponds to the mean values,  $\sigma$  to the standard deviation and  $M$  to the best solution found in each case. **BKS** indicates the best known solution for each test problem. We show in **boldface** those cases in which an approach reached the best known solution for that particular test problem.

	BKS	CDE			R. descent		
		$\mu$	$\sigma$	$M$	$\mu$	$\sigma$	$M$
<b>g01</b>	-15.000	-13.93461	$1.23 \times 10^0$	-14.99999	-11.25425	$2.64 \times 10^0$	-12.000
<b>g02</b>	-0.803619	-0.8033878	$6.83 \times 10^{-5}$	-0.8033884	-0.784641	$3.27 \times 10^{-2}$	-0.7930787
<b>g03</b>	-1.000	0.24616	$9.50 \times 10^{-2}$	-1.000	<b>-1.000</b>	$5.13 \times 10^{-9}$	-1.000
<b>g04</b>	-30665.539	<b>-30665.539</b>	$0.00 \times 10^0$	-30665.539	<b>-30665.539</b>	$0.00 \times 10^0$	-30665.539
<b>g05</b>	5126.498	5315.60	$3.01 \times 10^2$	5126.498	5195.899	$2.20 \times 10^2$	5126.500
<b>g06</b>	-6961.814	<b>-6961.814</b>	$0.00 \times 10^0$	-6961.814	-7229.388	$1.41 \times 10^3$	-7818.326
<b>g07</b>	24.3062091	24.78596	$3.12 \times 10^{-1}$	24.3062091	24.31361	$6.98 \times 10^{-2}$	24.30623
<b>g08</b>	-0.095825	-0.09583	$1.11 \times 10^{-3}$	-0.095826	<b>-0.095825</b>	$0.00 \times 10^0$	-0.095825
<b>g09</b>	680.6301	<b>680.630</b>	$0.00 \times 10^0$	680.630	<b>680.6301</b>	$0.00 \times 10^0$	680.6301
<b>g10</b>	7049.250	7090.50762	$3.99 \times 10^2$	7085.876	7094.163	$8.22 \times 10^1$	7049.396
<b>g11</b>	0.750	<b>0.750</b>	$0.00 \times 10^0$	0.750	0.9505711	$9.51 \times 10^{-2}$	1.000
<b>g12</b>	-1.000	<b>-1.000</b>	$0.00 \times 10^0$	-1.000	<b>-1.000</b>	$0.00 \times 10^0$	-1.000
<b>g13</b>	0.053950	0.80852	$1.87 \times 10^{-1}$	0.2476	0.3678503	$1.64 \times 10^{-1}$	0.4394295

**Table 3** Statistics with respect to the  $f(\mathbf{x})$  values obtained for a random choice of  $Cr$  and for **R1**.  $\mu$  corresponds to the mean values,  $\sigma$  to the standard deviation and  $M$  to the best solution found in each case. **BKS** indicates the best known solution for each test problem. We show in **boldface** those cases in which an approach reached the best known solution for that particular test problem.

	BKS	$Cr$ random			R1		
		$\mu$	$\sigma$	$M$	$\mu$	$\sigma$	$M$
<b>g01</b>	-15.000	-11.82917	$2.72 \times 10^0$	-12.000	-14.81745	$2.72 \times 10^0$	-15.000
<b>g02</b>	-0.803619	-0.8009026	$6.16 \times 10^{-3}$	-0.8036145	-0.8015735	$4.63 \times 10^{-3}$	-0.803613
<b>g03</b>	-1.000	<b>-0.9999999</b>	$3.09 \times 10^{-8}$	-0.9999999	<b>-0.9999997</b>	$7.55 \times 10^{-7}$	-0.9999999
<b>g04</b>	-30665.539	<b>-30665.539</b>	$0.00 \times 10^0$	-30665.539	<b>-30665.539</b>	$0.00 \times 10^0$	-30665.539
<b>g05</b>	5126.498	5171.077	$1.42 \times 10^2$	5126.498	5170.813	$1.42 \times 10^2$	5126.498
<b>g06</b>	-6961.814	<b>-6961.814</b>	$0.00 \times 10^0$	-6961.814	<b>-6961.814</b>	$0.00 \times 10^0$	-6961.814
<b>g07</b>	24.3062091	24.30722	$7.70 \times 10^{-3}$	24.306210	24.30708	$3.42 \times 10^{-4}$	24.30705
<b>g08</b>	-0.095825	<b>-0.095825</b>	$0.00 \times 10^0$	-0.095825	<b>-0.095825</b>	$0.00 \times 10^0$	-0.095825
<b>g09</b>	680.6301	<b>680.6301</b>	$0.00 \times 10^0$	680.6301	<b>680.6301</b>	$0.00 \times 10^0$	680.6301
<b>g10</b>	7049.250	7103.163	$8.81 \times 10^1$	7049.63	7104.965	$8.83 \times 10^1$	7049.783
<b>g11</b>	0.750	0.8992142	$1.19 \times 10^{-1}$	1.000	0.901635	$1.18 \times 10^{-1}$	1.000
<b>g12</b>	-1.000	<b>-1.000</b>	$0.00 \times 10^0$	-1.000	<b>-1.000</b>	$0.00 \times 10^0$	-1.000
<b>g13</b>	0.053950	0.3227286	$2.22 \times 10^{-1}$	0.4388694	0.3071761	$1.98 \times 10^{-1}$	0.4388515

On the other hand, Table 5 shows that the selection mechanism based on a roulette-wheel with random initial position (**R2**) required a lower number of iterations than the others in ten of the thirteen test problems adopted. This mechanism also had the lowest average number of generations in seven of the test problems adopted. This confirms that this selection mechanism had the best overall performance from all the approaches that were compared in our experimental study.

**Table 4** Statistics with respect to the  $f(x)$  values obtained for **R2** and **R3**.  $\mu$  corresponds to the mean values,  $\sigma$  to the standard deviation and  $M$  to the best solution found in each case. **BKS** indicates the best known solution for each test problem. We show in **boldface** those cases in which an approach reached the best known solution for that particular test problem.

	BKS	R2			R3		
		$\mu$	$\sigma$	$M$	$\mu$	$\sigma$	$M$
g01	-15.000	<b>-15.000</b>	$0.00 \times 10^0$	-15.000	-14.8200	$2.72 \times 10^0$	-15.000
g02	-0.803619	-0.8014436	$4.67 \times 10^{-3}$	-0.8014834	-0.8014504	$5.91 \times 10^{-3}$	-0.8036136
g03	-1.000	<b>-0.9999997</b>	$7.05 \times 10^{-7}$	-0.9999999	<b>-0.9999996</b>	$1.04 \times 10^{-6}$	-0.9999999
g04	-30665.539	<b>-30665.539</b>	$0.00 \times 10^0$	-30665.539	<b>-30665.539</b>	$0.00 \times 10^0$	-30665.539
g05	5126.498	5171.097	$1.42 \times 10^2$	5126.498	5171.311	$1.43 \times 10^1$	5170.288
g06	-6961.814	<b>-6961.814</b>	$0.00 \times 10^0$	-6961.814	<b>-6961.814</b>	$0.00 \times 10^0$	-6961.814
g07	24.3062091	<b>24.30649</b>	$7.54 \times 10^{-4}$	24.30627	24.30705	$3.31 \times 10^{-3}$	24.307
g08	-0.095825	<b>-0.095825</b>	$0.00 \times 10^0$	-0.095825	<b>-0.095825</b>	$0.00 \times 10^0$	-0.095825
g09	680.6301	<b>680.6301</b>	$0.00 \times 10^0$	680.6301	<b>680.6301</b>	$1.00 \times 10^{-5}$	680.6301
g10	7049.250	7103.024	$8.82 \times 10^2$	7049.556	7106.478	$8.92 \times 10^1$	7049.664
g11	0.750	0.8997895	$1.18 \times 10^{-1}$	1.000	0.9016549	$1.18 \times 10^{-1}$	1.000
g12	-1.000	<b>-1.000</b>	$0.00 \times 10^0$	-1.000	<b>-1.000</b>	$0.00 \times 10^0$	-1.000
g13	0.053950	0.3132858	$2.13 \times 10^{-1}$	0.4388565	0.3144292	$2.11 \times 10^{-1}$	0.4388585

**Table 5** Average and minimum number of generations required for each algorithm to reach the best known solution to each of the test problems adopted.

	CDE		R. Desc.		Cr random		R1		R2		R3	
	$\mu$	min	$\mu$	min	$\mu$	min	$\mu$	min	$\mu$	min	$\mu$	min
g01	614.88	543	725.93	714	727.84	563	722.37	763	<b>549.32</b>	<b>541</b>	720.14	701
g02	<b>5843.77</b>	5954	5963.33	5934	5960.73	5979	5939.29	5954	5950.26	<b>5907</b>	5975.90	5995
g03	<b>4134.24</b>	3152	5719.55	4217	5693.44	4218	5650.55	4174	5682.42	<b>3128</b>	5598.37	3505
g04	675.33	476	872.34	758	887.19	738	867.98	749	<b>510.26</b>	<b>449</b>	868.86	719
g05	<b>3456.54</b>	1652	4395.93	1786	4293.48	3059	4448.92	1411	3855.19	<b>1346</b>	4316.96	1613
g06	1162.60	<b>154</b>	1100.83	435	<b>931.24</b>	423	1092.50	445	1070.48	291	965.39	496
g07	5945.12	<b>4130</b>	5973.51	5726	<b>5762.79</b>	5834	5766.33	5730	5903.23	4976	6783.88	5757
g08	165.75	98	214.59	100	163.16	97	161.65	97	<b>161.61</b>	<b>78</b>	162.18	102
g09	2621.86	1068	3878.70	1481	3520.84	1581	3518.51	1374	<b>2393.24</b>	<b>1028</b>	3417.16	1384
g10	5954.63	5921	5920.25	5975	5921.15	5985	5943.33	<b>4895</b>	<b>5875.74</b>	4987	5907.12	4989
g11	536.14	367	457.28	302	401.55	302	417.25	302	<b>166.14</b>	<b>230</b>	391.43	466
g12	201.72	163	158.95	134	163.48	119	161.76	129	<b>159.32</b>	<b>119</b>	172.54	132
g13	<b>4119.88</b>	4003	4577.57	3876	5184.46	2529	5007.46	3906	5208.23	<b>2258</b>	5136.14	3111

## 7 Conclusions and Future Work

We have proposed here a new hyper-heuristic for solving constrained optimization problems. The proposed approach uses as its low-level heuristics a set of twelve differential evolution variants. Additionally, the selection mechanism of differential evolution was modified in order to make it able to handle constraints (stochastic ranking was adopted for this sake).

The results obtained by our proposed approach are very promising, since they are better than those produced by a state-of-the-art DE-based evolutionary optimization approach (CDE). This indicates that the mechanism adopted by our hyper-heuristic is working in a proper way.



As part of our future work, we aim to improve the selection mechanism of our hyper-heuristic. In order to achieve that, it is required to perform a more in-depth study of the different DE variants adopted here, so that we can understand in a better way how they work when dealing with constrained optimization problems. We are also interested in adding to our hyper-heuristic other low-level heuristics such as particle swarm optimization [9] and evolution strategies [13], since we believe that such approaches perform search movements that could complement those produced by differential evolution. Evidently, the goal of adding more heuristics would be to improve the performance of our hyper-heuristic.

**Acknowledgements.** The second author acknowledges support from CONACyT project 103570.

## References

1. Biazzi, M., Bánhelyi, B., Montresor, A., Jelasity, M.: Distributed hyper-heuristics for real parameter optimization. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 1339–1346. ACM, Montréal Québec (2009)
2. Burke, E., Hart, E., Kendall, G., Newall, J.: Hyper-Heuristics: An Emerging Direction In Modern Search Technology, handbook of metaheuristics edn., ch. 16, pp. 457–474. Springer, New York (2003)
3. Burke, E., Kendall, G., Soubeiga, E.: A tabu-search hyper-heuristic for timetabling and rostering. *Journal of Heuristics* 9(6), 451–470 (2004)
4. Chakhlevitch, K., Cowling, P.: Hyperheuristics: Recent Developments. *SCI*, vol. 136, pp. 3–29. Springer, Berlin (2008)
5. Cowling, P.I., Kendall, G., Soubeiga, E.: A Hyperheuristic Approach to Scheduling a Sales Summit. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 176–190. Springer, Heidelberg (2001)
6. De Jong, K.A.: An analysis of the behaviour of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan (1975)
7. Fan, Z., Liu, J., Sorensen, T., Wang, P.: Improved differential evolution based on stochastic ranking for robust layout synthesis of mems components. *IEEE Transactions On Industrial Electronics* 56(4), 937–948 (2008)
8. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading (1989)
9. Kennedy, J., Eberhart, R.C.: *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco (2001)
10. Lampinen, J.: Constraint handling approach for the differential evolution algorithm. In: Proceedings of the Congress on Evolutionary Computation 2002 (CEC 2002), vol. 2, pp. 1468–1473. IEEE Service Center, Piscataway (2002)
11. Price, K.: An introduction to differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimization*, pp. 79–106. McGraw-Hill (1999)
12. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. *Transactions On Evolutionary Computation* 4(3), 284–294 (2000)
13. Schwefel, H.P.: *Evolution and Optimum Seeking*. John Wiley & Sons, New York (1995)

14. Storer, R., Wu, S., Vaccari, R.: Problem and heuristic search space strategies for job shop scheduling. *ORSA Journal on Computing* 7, 453–467 (1995)
15. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Tech. Rep. TR-95-012, International Computer Science Institute (1995)
16. Wolpert, D., MacReady, W.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1), 67–82 (1997)

**Part VII**  
**Evolutionary Computation for Vision,  
Graphics, and Robotics**

# Evolutionary Computation Applied to the Automatic Design of Artificial Neural Networks and Associative Memories

Humberto Sossa, Beatriz A. Garro, Juan Villegas, Gustavo Olague,  
and Carlos Avilés

**Abstract.** In this paper we describe how evolutionary computation can be used to automatically design artificial neural networks (ANNs) and associative memories (AMs). In the case of ANNs, Particle Swarm Optimization (PSO), Differential Evolution (DE), and Artificial Bee Colony (ABC) algorithms are used, while Genetic Programming is adopted for AMs. The derived ANNs and AMs are tested with several examples of well-known databases.

## 1 Introduction

If we want that a machine efficiently interacts with its environment, it is necessary that the so called pattern recognition problem is appropriately solved. Lots of approaches to face this problem have been reported in literature. One of the most popular one is the artificial neural network based approach. It consists on combining the individual capacities of many small processors (programs) in such a way that a set of patterns under study is correctly classified or restored.

An artificial neural network (ANN) can be seen as a set of highly interconnected processors. The processors can be electronic devices or computer programs. From

---

Humberto Sossa · Beatriz A. Garro

CIC-IPN, Juan de Dios Batiz S/N, Col. Nva. Industrial Vallejo, Mexico City, Mexico  
e-mail: [hsossa@cic.ipn.mx](mailto:hsossa@cic.ipn.mx), [beatriz.auroragl@gmail.com](mailto:beatriz.auroragl@gmail.com)

Juan Villegas · Carlos Avilés

UAM-Azcapotzalco, Av. San Pablo Xalpa 180. Azcapotzalco, Mexico City, Mexico  
e-mail: [jvillegas@gmail.com](mailto:jvillegas@gmail.com), [caviles@correo.azc.uam.mx](mailto:caviles@correo.azc.uam.mx)

Gustavo Olague

CICESE, Carretera Ensenada-Tijuana 3918 Zona Playitas, Ensenada, B. C., Mexico  
e-mail: [gustavo.olague@gmail.com](mailto:gustavo.olague@gmail.com)

now on, these processors will be called nodes or units. These units can be the nodes of a graph. The edges of this graph determine the interconnections among the nodes. These represent the synaptic connections between the nodes, and are supposed to be similar to the synaptic connections between biological neurons of a brain.

Associative memories, in the other hand, are special cases of ANNs. They have several interesting properties that make them preferable than ANNs, for some problems.

In this paper, we briefly describe how bio-inspired and evolutionary based techniques can be efficiently used for the automatic design of ANNs and AMs. The rest of the paper is organized as follows. Section 2 is focused to explain the generalities of ANNs and AMs. Section 3 is oriented to explain the generalities about how three bio-inspired techniques: Particle Swarm Intelligence (PSO), Differential Evolution (DE) and Artificial Bee Colony (ABC) have been used with success in the design of ANNs to classify patterns. Section 4, in the other hand, is dedicated to provide the details of how Genetic Programming can be used to synthesize AMs for pattern classification as well as pattern restoration. Section 5 is devoted to present some of the obtained results. A discussion of the results is also given in this section. Finally, Section 6 is oriented for the conclusions and directions for further research.

## 2 Basics on Artificial Neural Networks and Associative Memories

In this section we present the most relevant concepts and definitions concerning artificial neural networks.

### 2.1 Basics on Artificial Neural Networks (ANNs)

An ANN is an interconnected set of simple processing elements, units or nodes, whose functionality is vaguely based on the animal neuron. The processing ability of the net is stoked in the connections (weights) among the units. These values are obtained by means of an adapting or learning process from a learning set [7].

An ANN performs a mapping between the input vector  $X$  and an output vector  $Y$ , by the consecutive application of two operations. The first operation computes at each node the alignment between the input vector  $X$  and the auxiliary weighting vector  $W$ . The second operation takes the result of the first operation and computes the mapping. The two equations that govern the functionality of an individual processing unit  $j$  net without bias  $b_j$  are the following:

$$a = \sum_{i=1}^n w_i x_i, \quad (1)$$

$$y_j = f(a), \quad (2)$$

where the  $x_i$  are the inputs to the neuron and the  $w_i$  are its synaptic weights. In matrix form:

$$a = W \cdot X, \quad (3)$$

where now  $W = (w_1, w_2, \dots, w_n)^T$  and  $X = (x_1, x_2, \dots, x_n)^T$ .

As we can see the neuron performs the dot product between the weight vector  $W$  and the input vector  $X$ . The output function  $f(a)$  of the neuron is usually non-linear. In the case of the threshold logic unit proposed by McCulloch and Pitts [15] is the hard limit function or in the case of the Perceptron [18] is the sigmoid function [7], [19].

The way the set of neurons are interconnected determines the ANN architecture. The neurons in an ANN can be connected feedforward, sometimes they can admit side connections, even feedback.

Three elements characterize the functionality of an ANN: its architecture (the way its nodes are interconnected), the values of its weights, and the transfer functions that determine the kind of output of the net.

In Section 3 we will see how for a given ANN to automatically select each of these components, and this by means of bio-inspired techniques.

## 2.2 Basics on Associative Memories (AMs)

An AM is a mapping used to associate patterns from two different spaces. Mathematically, an AM,  $M$  is a mapping that allows restoring or recalling a pattern  $y^k$ ,  $k = 1, \dots, p$ , given an input pattern  $x^k$ ,  $k = 1, \dots, p$ . In general  $y^k$  is of dimension  $m$ , while  $x^k$  is of dimension  $n$ .

Both  $x^k$  and  $y^k$  can be seen as vectors as follows:  $x^k = (x_1^k, \dots, x_n^k)^T$  and  $y^k = (y_1^k, \dots, y_m^k)^T$ . Thus:

$$x^k \rightarrow M \rightarrow y^k. \quad (4)$$

If for all  $k$ ,  $x^k = y^k$ , the memory operates in auto-associative way, otherwise it works as a hetero-associative operator. For each  $k$ ,  $(x^k, y^k)_{k=1}^p$  is called an association. The whole set of associations is called the fundamental set of associations.

Examples of AMs are the Linear Associator (LA) [1]-[10], the Lernmatrix (LM) [20], and the morphological associative memory (MAM) [17]. Both the LA and the LM operate in the hetero-associative way, while the MAM can operate in auto and hetero-associative fashions. The LA and the LM operate with binary-valued vectors. MAMs can operate both with binary or real-valued vectors.

To operate an AM two phases are required, one of construction or designing and one of testing or retrieval.

Generally, in the design of an AM two operators are required: an internal operator and an external operator. Internal operator  $O_I$  is used to derive a partial codification of the set of patterns. It acts on each association:  $(x^k, y^k)_{k=1}^p$ . It gives, as a result a part of the AM. External operator  $O_E$ , on the other hand, combines the partial results obtained by  $O_I$ , and gives, as a result, the total mapping  $M$ .

As an illustrative example, let us take the case of the LA. In a first step, the LA takes each association  $(x^k, y^k)_{k=1}^p$  and produces the partial codification:

$$M^k = y^k (x^k)^T \quad (5)$$

It then takes the  $k$  partial matrices and produces the final mapping:

$$M = M^1 + \dots + M^p = \sum_{k=1}^p y^k (x^k)^T. \quad (6)$$

As can be seen from this example, the following two operations are needed to get the LA: a product between each two vectors:  $y^k$  and  $x^k$ , to get matrix  $M^k$ , and a sum between matrices  $M^k$  to the final  $M$ .

Recalling of a given pattern  $y^k$ , through a designed  $M$  is given as follows:

$$y^k = M \cdot x^k. \quad (7)$$

In this case, recalling demands only one operation, a multiplication between the LA and the input vector.

Necessary conditions for correct recall of each  $y^k$  is that the all the  $x^k$  are orthonormal. This is a very restrictive condition but it allows visualizing the necessary operations to operate the memory.

### 3 Automatic Synthesis of ANNs

The designing of an ANN normally involves 1) the automatic adjustment of the synaptic weights between the different neurons of the ANN, 2) the selection of the corresponding architecture of the ANN, and 3) the selection also of the transfer function of the neurons is also, sometimes, a matter.

Several methods to adjust the weights of the ANN, once its architecture has been selected, can be found in the literature. Probably, the most known in the case of arrangement of Perceptrons is the back-propagation rule (BP) [19]. It is based on gradient decent principle and, if no convenient actions are taken into account BP, generally falls into a local minimum providing a non optimal solution. Since the point of view of pattern classification this could be interpreted as a deficient learning of the ANN, and/or a bad generalization capacity.

Since several years many scientists have used evolutionary and bio-inspired techniques to evolve: 1) the synaptic weights of the ANN, 2) its architecture, or 3) both the synaptic weights and its architecture. For a good review on the subject refer, for example, to [28]-[29]. In [2], [3], [4], [5] and [6], the authors show how bio-inspired techniques such as PSO, DE and ABC can be used to automatically select the architecture of and ANN, tune its weights and even to chose the transfer function for each neuron. In this section we give the generalities of this proposal. Related work concerning the training of spiking neurons by means bio-inspired techniques can be found in [22], [23], [24] and [25].

### 3.1 PSO, DE and ABC

PSO, DE and ABC are examples of searching techniques to solve optimization problems with many optima where most standard methods will fail. Generally speaking, a bio-inspired technique is a method inspired in a metaphor of nature that takes into account the partial but powerful abilities of each of the individuals to produce a global solution for a difficult problem. A typical problem to solve is searching for food. For example, the individuals of the colony of ants experiment hanger but they do not know where the food is. A subset of the ants goes in all directions inside their territory searching for the precious product. Ants communicate among themselves by so-called pheromones. Once an ant or a group of ants find the desired food, they communicate to the other. The information goes back as a chain to the nest, the collector ants then go for the food. Details about the functioning of PSO, DE and ABC can be found in [9], [21], and [8], respectively.

### 3.2 Garro’s Proposal

The problem to be solved is stated as follows:

Given a set of input patterns  $X = \{X^1, \dots, X^p\}$ ,  $X^k \in \mathbb{R}^n$ , and a set of desired patterns  $D = \{d^1, \dots, d^p\}$ ,  $d^k \in \mathbb{R}^m$ , find an ANN represented by a matrix  $W \in \mathbb{R}^{q \times (q+1)}$ , such that a function defined as  $\min(f(X, D, W))$  is minimized. In this case  $q$  is the maximum number of neurons MNN; it is defined as  $q = 2(m + n)$ .

Each individual ANN is codified as a matrix as follows:

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,NMN+1} & x_{1,NMN+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{NMN,1} & x_{NMN,2} & \dots & x_{NMN,NMN+1} & x_{NMN,NMN+2} \end{bmatrix} \quad (8)$$

The matrix is composed by three parts: The topology (the first column of (8)), the transfer functions (the last column of (8)), and the synaptic weights (the submatrix of (8) without the first and last column).



The aptitude of an individual is computed by means of the MSE function:

$$F1 = \frac{1}{p \cdot m} \sum_{\xi=1}^p \sum_{j=1}^m (d_j^{\xi} - y_j^{\xi})^2. \quad (9)$$

This way, all the values of matrix  $W$  are codified so as to obtain the desired ANN. Moreover, each solution must be tested in order to evaluate its performance. For this, it is necessary to know the classification error (CER), this is to know how many patterns have been correctly classified and how many were incorrectly classified. Based on the winner-take-all technique the CER function can be computed as follows:

$$F2 = 1 - \frac{nwcp}{tnp} \quad (10)$$

In this case,  $nwcp$  is the number of well classified patterns and  $tnp$  is the total number of patterns to be classified.

Additionally, if we want to minimize the number of connections of the ANN, we would also make use of the following function:

$$F3 = \frac{NC}{NmaxC}. \quad (11)$$

In this case  $NC$  is the number of connections of the ANN, while  $NmaxC = \sum_{i=n}^{MNN} i$  is the maximum number of connections generated with  $MNN$  neurons. When functions  $F1$ ,  $F2$  and  $F3$  are combined, we get the two functions to be optimized:

$$FF1 = F1 \cdot F2 \quad (12)$$

$$FF2 = F1 \cdot F3 \quad (13)$$

The six transfer functions used by Garro are the logsig (LS), tansig (TS), sin (S), radbas (RD), pureline (PL), and hardlim (HL). These functions were selected for they are the most popular and useful transfer functions in several kinds of problems.

In Section 5, we will see how these two functions can be used to automatically synthesize an ANN for a given classification problem.

## 4 Automatic Synthesis of AMs by Means of Genetic Programming

Until 2005 all the AMs models found in literature (more or less 30) have been produced by a human user. In 2009, in [26] and [27], the authors arrive to an original solution where, for the first time, they propose a methodology for the automatic synthesis of AMs for pattern restoration. In this section we provide the generalities of this proposal.

## 4.1 Genetic Programming

Genetic programming (GP) as proposed by J. R. Koza is an evolutionary algorithm-based methodology inspired by biological evolution to find computer programs that perform a user-defined task [11], [12], [13] and [14]. It is a specialization of genetic algorithms (GA) where each individual is a computer program. Therefore, GP is a machine learning technique used to optimize a population of computer programs according to a fitness function determined by a program's ability to perform a given computational task. The idea behind GP is to evolve computer programs represented in memory as tree structures. Basically, the way to modify the trees is carried out in two ways, either by crossing them or mutating them. This way we can evaluate the performance of the trees. At the end of the process we will have the winner tree or winner trees. To generate a solution, GP operates onto two sets (a terminal set, TS, and a function set, FS) and a fitness function FF. At the end of the evolving process GP delivers one or more solutions as programs that solve the problem.

## 4.2 Villegas' Proposal

We have seen that to operate an AM two operators are required, one for designing the memory and for testing the memory. Let us design these operators as follows:  $D_O$ , for designing operator or codifying operator and  $D_T$ , for testing operator.

The general idea of the technique proposed by Villegas to automatically synthesize an AM by means of genetic programming is as follows, given a set of associations  $(x^k, y^k)_{k=1}^p$ :

1. Propose a set of initial  $Q$  solutions, this a set of couples of operators:  $(D_O, D_T)^q$ ,  $q = 1, \dots, Q$ , each one expressed as a tree in terms of the chosen function and terminal sets,  $F$  and  $T$ .
2. Test the different couples  $(D_O, D_T)^q$  with the  $p$  associations  $(x^k, y^k)_{k=1}^p$ , and retain the best solutions according the chosen fitness function  $FF$ .
3. Evolve the couples.
4. Repeat Steps 2 and 3 until obtaining the set of the best solutions.

The result is a set of several evolved couples  $(D_O^*, D_T^*)$  that best satisfy fitness function  $FF$ . In the next section we will show examples of obtained couples for several pattern restoration examples.

## 5 Experimental Results

Here, we present several examples of the ANNs and AMs automatically obtained by the proposals explained in Sections 3 and 4. We provide also a discussion to complete explanation.

## 5.1 Examples of Synthetically Generated ANNs

The methodology described in Section 3 was applied to several well-known pattern recognition problems. The following pattern classification problems taken from the machine learning benchmark repository UCI were taken [16]: iris plant database, wine database and breast cancer database. Due to space limitations, we only show results concerning the application of ABC technique to the iris plant database. The iris plant database consists of 150 samples, described by four features: 1) length of the sepal, 2) width of the sepal, 3) length of the petal, and 4) width of the petal, all in cm. Ten experiments were performed for each of the three databases.

Figure 1 shows the evolution of the error for functions and for the iris database. Figure 2(a) shows one of the ANNs obtained by the proposed methodology. Figure 2(b) shows one of the ANNs obtained by the proposed methodology taking into account  $F3$ . Note the reduction of the connections. Figure 3 shows the percentages of recognition for the ten experiments. Note that in all ten experiments, the percentage of recognition maintains high.

Note also how reducing the number of connections does not dramatically affect the performance of the synthesized ANN. From this experiment we can conclude that the proposed methodology provides very promising results.

## 5.2 Examples of Synthetically Generated AMs

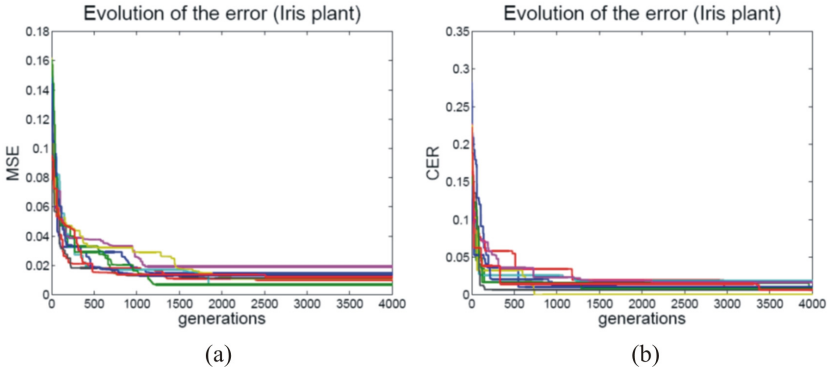
The methodology described in Section 4 was applied to several well-known pattern recognition problems. According to [26] and [27]:

For association:

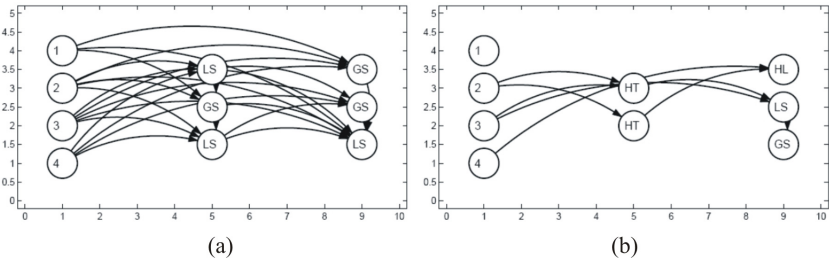
- $O_{p_k^a}$  is the evolved operator for pattern association.
- $M^k$  is the partial association matrix by applying operator  $O_{p_k^a}$  to each association  $(x^k, y^k)$ .
- $M$  is the associative memory that comes up from the addition of all the  $M^k$ .
- $TS_a = \{x^k, y^k\}$  is the set of terminals for association.
- $FS_a = \{+, -, \min, \max, times\}$  is the set of functions for association.

For recalling:

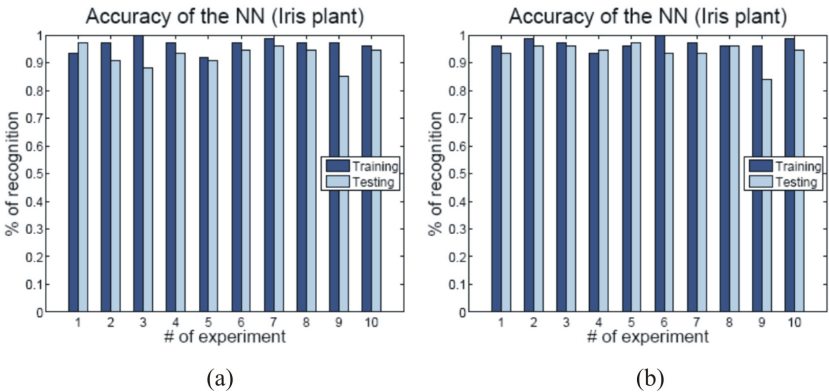
- $O_{p_k^r}$  is the evolved operator for pattern recalling or classification.
- $TS_r = \{v, row1, row2, \dots, rowm, M^k\}$  is the set of terminals for pattern recalling or classification.  $v$  is the input vector,  $rowj$  is the  $j$ -th row of matrix  $M$ .
- $FS_a = \{+, -, \min, \max, mytimesm\}$  is the set of functions for pattern recalling or classification.
- $mytimesm$  operation produces a vector and is defined as:  $mytimesm(x^k, y^k) = [x_1y_1, \dots, x_ny_n]$ .
- $\hat{y}^k$  is the recalled by applying operator  $O_{p_k^r}$  to input vector  $x(\tilde{x})$ .  $\tilde{x}$  is a distorted version of  $x$ .



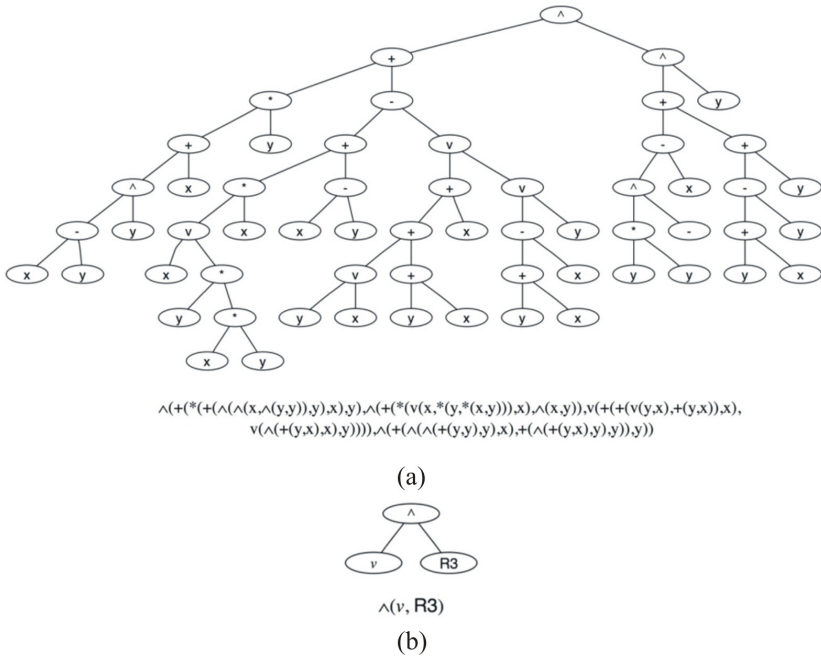
**Fig. 1** Evolution of the error for the ten experiments for the Iris plant problem. (a) Evolution of  $FF1$  using MSE function. (b) Evolution of  $FF2$  using CER function. Figure taken from [5].



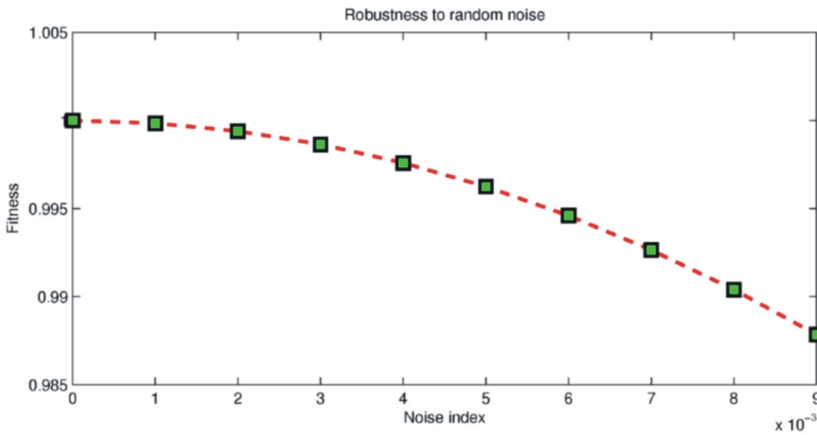
**Fig. 2** Two different ANNs designs for the Iris plant problem. (a) ANN designed by the ABC algorithm without taking into account  $F3$  function. (b) ANN designed by the ABC algorithm taking into account  $F3$  function. Figure taken from [5].



**Fig. 3** Percentage of recognition for the Iris problem and the ten experiments during the training and testing stage for each fitness function. (a) Percentage of recognition minimizing the  $FF1$  function. (b) Percentage of recognition minimizing the  $FF2$  function. Figure taken from [5].



**Fig. 4** A couple of association and recalling operators automatically derived by the proposal for the Iris database problem. Taken from [27].



**Fig. 5** Original images of the first ten digits and noisy versions of them, used to test the proposal. Taken from [27].

The fitness function used to test the efficiency of the evolved operators is:

$$f = \frac{y\bar{y}}{\sqrt{y\bar{y}} \cdot \sqrt{y\bar{y}}} \quad (14)$$

The proposed methodology was tested with several pattern classification examples: Database of numbers and the same databases used in Section 3. Due to space limitations, only results with the Iris Plant classification database already used in Section 3.

We implemented our model looking for several pairs of association and recalling operators. One of these pairs is shown in Fig. 4. We tested this AM by adding random noise to the input pattern set as shown in Fig 5. In this case the noise was added from 0.01 to 0.09% in steps of 0.01. While the fundamental set was correctly recalled, the recalling rate decreased slowly as noise increased. One can note the complexity of the operator for association, compared with other associations proposed by humans. Note however the simplicity of the recalling operator for this example.

## 6 Conclusions and Directions for Further Research

We have seen that it is possible to automatically synthesize ANNs and AMs for pattern classification and pattern restoration porpoises. In the case of ANNs we have made use of bio-inspired techniques such as PSO, DE and ABC; while for AMs we have used GP. In both cases we very nice and promising results have been obtained. We have tested the proposed techniques with several reported benchmarks with satisfactory results.

Venues for further research are the following: 1) Automatic design of radial base networks, 2) Automatic design of morphological neural networks. 3) Automatic design of spiking neural networks. 4) Automatic design of bidirectional associative memories, and 5) Simplification of associative memory operators.

**Acknowledgements.** H. Sossa thanks SIP-IPN and CONACYT for the economical support under grants SIP 20111016, SIP 20121311 and CONACYT 155014 to develop the reported investigations. We also thank CIC-IPN, CICESE and UAM-Azcapotzalco for the support to undertake this research.

## References

1. Anderson, J.A.: A simple neural network generating an interactive memory. *Mathematical Biosciences* 14, 197–220 (1972)
2. Garro, B.A., Sossa, H., Vázquez, R.A.: Design of Artificial Neural Networks using a Modified Particle Swarm Optimization Algorithm. In: *International Joint Conference on Neural Networks (IJCNN 2009)*, Atlanta, GE, USA, June 14-19, pp. 938–945 (2009)

3. Garro, B.A., Sossa, H., Vázquez, R.A.: Design of Artificial Neural Networks Using Differential Evolution Algorithm. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) *ICONIP 2010, Part II. LNCS*, vol. 6444, pp. 201–208. Springer, Heidelberg (2010)
4. Garro, B.A., Sossa, H., Vázquez, R.A.: Evolving Neural Networks: A Comparison between Differential Evolution and Particle Swarm Optimization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) *ICSI 2011, Part I. LNCS*, vol. 6728, pp. 447–454. Springer, Heidelberg (2011)
5. Garro, B.A., Sossa, H., Vázquez, R.A.: Artificial Neural Network Synthesis by means of Artificial Bee Colony (ABC) Algorithm. In: *CEC 2011, New Orleans, June 5-8 (2011)*
6. Garro, B.A., Sossa, H., Vázquez, R.A.: Back-Propagation vs Particle Swarm Optimization Algorithm: which Algorithm is better to adjust the Synaptic Weights of a Feed-Forward ANN? *International Journal of Artificial Intelligence* 7(11), 208–218 (2011)
7. Gurney, K.: *An Introduction to Neural Networks*. Taylor and Francis Group (1997)
8. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. *Journal of Global Optimization* 39(3), 459–471 (2007)
9. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, vol. IV, pp. 1942–1948 (1995)
10. Kohonen, T.: Correlation matrix memories. *IEEE Transactions on Computers* C-21(4), 353–359 (1972)
11. Koza, J.R.: *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Stanford University Computer Science Department technical report (1990)
12. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press (1992)
13. Koza, J.R.: *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press (1994)
14. Koza, J.R., Bennett, F.H., Andre, D., Keane, M.A.: *Genetic Programming III: Darwinian Invention and Problem Solving*. Morgan Kaufmann (1999)
15. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 7, 115–133 (1943)
16. Murphy, P.M., Aha, D.W.: *UCI Repository of machine learning databases*. University of California, Department of Information and Computer Science, Irvine, CA, US., Technical Report (1994)
17. Ritter, G., Diaz, J.: Morphological associative memories. *IEEE Transactions on Neural Networks* 9(2), 281–293 (1998)
18. Rosenblatt, F.: *The Perceptron—a perceiving and recognizing automaton*. Report 85-460-1, Cornell Aeronautical Laboratory (1957)
19. Rojas, R.: *Neural networks - A systematic introduction*. Chapter 7: The back propagation algorithm (1996)
20. Steinbuch, K.: Die Lernmatrix. *Kybernetik* 1(1), 36–45 (1961)
21. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341–359 (1997)
22. Vazquez, R.A.: Izhikevich Neuron Model and its Application in Pattern Recognition. *Australian Journal of Intelligent Information Processing Systems* 11(1), 53–60 (2010)
23. Vázquez, R.A.: Pattern Recognition Using Spiking Neurons and Firing Rates. In: Kuri-Morales, A., Simari, G.R. (eds.) *IBERAMIA 2010. LNCS*, vol. 6433, pp. 423–432. Springer, Heidelberg (2010)
24. Vazquez, R.A.: A computational approach for modelling the biological olfactory system during an odour discrimination task using spiking neuron. *BMC Neuroscience* 12(supp.1), 360 (2011)

25. Vázquez, R.A., Garro, B.A.: Training Spiking Neurons by Means of Particle Swarm Optimization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part I. LNCS, vol. 6728, pp. 242–249. Springer, Heidelberg (2011)
26. Villegas, J., Sossa, H., Avilés, C., Olague, G.: Automatic Synthesis of Associative Memories by genetic Programming, a First Approach. *Research in Computing Science* 42, 91–102 (2009)
27. Villegas, J., Sossa, H., Avilés, C., Olague, G.: Automatic Synthesis of Associative Memories through Genetic Programming: a co-evolutionary approach. *Revista Mexicana de Física* 57(2), 110–116 (2011)
28. Yao, X.: A review of evolutionary artificial neural networks. *Int. J. Intell. Syst.* 8(4), 539–567 (1993)
29. Yao, X.: Evolutionary artificial neural networks. In: Kent, A., Williams, J.G. (eds.) *Encyclopedia of Computer Science and Technology*, vol. 33, pp. 137–170. Marcel Dekker, New York (1995)



# Segmentation of Blood Cell Images Using Evolutionary Methods

Valentín Osuna, Erik Cuevas, and Humberto Sossa

**Abstract.** Acute lymphoblastic leukemia is a blood cancer that can be cured if it is detected at early stages; however, the analysis of smear blood by a human expert is tired and subject to errors. In such a sense, diagnostic of the disease is costly and time consuming. Considering that situation, several automatic segmentation methods have been proposed, some of them containing combinations of classic image analysis tools, as thresholding, morphology, color segmentation and active contours, only to mention some. In this paper is proposed the use of Hellinger distance as an alternative to Euclidean distance in order to estimate a Gaussian functions mixture that better fits a gray-level histogram of blood cell images. Two evolutionary methods (Differential Evolution and Artificial Bee Colony) are used to perform segmentation based on histogram information and an estimator of minimum distance. The mentioned techniques are compared with classic Otsu's method by using a qualitative measure of the resulting segmentation and ground-truth images. Experimental results show that the three methods performed almost in a similar fashion, but the evolutionary ones evaluate almost 75 % less the objective function compared with Otsu's. Also, was found that the use of a minimum distance estimator constructed with Hellinger distance and evolutionary techniques is robust and does not need a penalization factor as the needed when an Euclidean distance is used.

## 1 Introduction

As in other types of cancer, an early diagnostic of Leukemia is important to increase a person's cure possibilities. In order to obtain a reliable diagnostic, a human expert must obtain information from smear blood samples, such as counting the number

---

Valentín Osuna · Humberto Sossa

CIC-IPN, Juan de Dios Batiz S/N, Col. Nva. Industrial Vallejo, Mexico City, Mexico

e-mail: [valentin.osuna@cucei.udg.mx](mailto:valentin.osuna@cucei.udg.mx), [hsossa@cic.ipn.mx](mailto:hsossa@cic.ipn.mx)

Erik Cuevas

CUCEI-UDEG, Dept. de C. Comput, Av. Revolucin 1500, Guadalajara, Jal, Mexico

e-mail: [erik.cuevas@cucei.udg.mx](mailto:erik.cuevas@cucei.udg.mx)

of red blood cells as well as the number and type of white blood cells, even considering other kind of artifacts that could represent either some particular disease or individual condition; however, the analysis of smear blood by a human expert is tired and subject to errors. For that reason, research on automatic segmentation of blood cells has been growing over the past years.

In general terms, image segmentation divides an image into related sections or regions, consisting of image pixels having related data feature values. It is an essential issue since it is the first step for image understanding, and any other step, such as feature extraction and recognition, heavily depends on its results. Segmentation algorithms are based on two significant criteria [1]: the homogeneity of a region (thresholding) and the discontinuity between adjacent disjoint regions (finding edges). Since the segmented image obtained from the homogeneity criterion has the advantage of smaller storage space, fast processing speed and ease in manipulation, thresholding techniques are considered the most popular [9, 23].

Particularly in medical image analysis, a correct segmentation is the most important step to perform a fully automatic and robust vision system for detection and classification of blood cell, and in order to segmenting such kind of images there exist several proposals. For instance, a segmentation method based in K- means, fuzzy Gustafson Kessel and Rough Sets clustering [15, 16, 17] with classification of nearest neighbor in four classes and using a Lab color space are proposed. By using color models RGB and HSI, segmentation of white blood cell nucleus, or blasts, is performed in [8] and later, both segmentations according to color model are compared. In [7] a RGB image is enhanced transformed to HSI color space and the channel S is segmented using an experimental threshold value. Based on the idea that images have homogeneous regions, in [25] the so called Color-Structure-Code is used to generate an ordered graph. In most of the methods reported in literature, it is necessary to perform an enhancing pre- processing step before the segmentation, increasing therefore the processing time, even considering only segmentation.

On the other hand, several nature inspired algorithms have emerged since first evolutionary algorithms were proposed in the 60's, leaving clear that biological inspired methods can be successfully transferred into novel computational paradigms [26]. Even today, this trend is still valid, as is shown by the development and use of concepts such as artificial neural networks, evolutionary algorithms, swarming algorithms and so on. Particularly, in this work are presented Differential Evolution (DE) and Artificial Bee Colony Optimization (ABC) to perform multi-threshold segmentation. DE is a population-based algorithm in which the population is evolved from one generation to the next using special defined operators such as mutation, crossover, and selection. More recently in 2005, the ABC algorithm has been introduced by Karaboga [10]. Such algorithm, inspired by the intelligent behavior of honey-bees, consists of three essential components: food source positions, nectar-amount and several honey-bee classes. Each food source position represents a feasible solution for the problem under consideration. The nectar-amount for a food source represents the quality of such solution (represented by fitness value). Each bee-class symbolizes one particular operation for generating new candidate food source positions. The aforementioned algorithms have been used to deal with

several optimization problems in the area of image analysis, giving good results in terms of performance [25, 3, 27].

In this work, the segmentation approach is based on a parametric model composed by a group of Gaussian functions (Gaussian mixture). Gaussian mixture (GM) represents a flexible method of statistical modelling with a wide variety of scientific applications [13, 21]. In general, GM involves the model selection, i.e., to determine the number of components in the mixture (also called model order), and the estimation of the parameters of each component in the mixture that better adjust the statistical model. Computing the parameters of Gaussian mixtures is considered a difficult optimization task, sensible to the initialization [20, 18] and full of possible singularities [11, 12]. As an optimization problem, the presented here requires an objective function, which makes use of Hellinger distance to compare the GM candidate and the original histogram. This distance measure works with probability density functions, making it appropriate to the problem presented in this work, and was shown that this distance is the most suitable to construct a minimum distance estimator [5]. The Hellinger distance has been used in on-line recognition of handwritten text [14], in signal modulation [22] and classification and localization of underwater acoustic signals [2], only to mention some uses.

Also, in this paper are shown experimental statistical comparisons among the two evolutionary methods and Otsu's approach with a database smear blood cell images [4]; such comparisons were made according the number of function evaluations and a qualitative measure based on Hausdorff distance [23, 6]. The rest of the paper is organized as follows. Next section is devoted to explain Gaussian approximation of the histogram and the Otsu method. In section 3 we show a brief overview of Differential Evolution and Artificial Bee Colony optimization, respectively, as well as some of their implementation details. Section 4 is focused to present the experimental results and comparisons, while Section 5 is oriented to give the conclusions and directions for future work.

## 2 Gaussian Approximation and Otsu's Methods

In this section we briefly describe the adopted Gaussian approximation as well as the basics of the thresholding method to get a binary version of an image.

### 2.1 Gaussian Approximation Method

In this work, the histogram  $h(g)$  of an image with  $L$  gray levels and its normalized version are given as:

$$h(g) = \frac{n_g}{N}, \quad h(g) \geq 0, \quad (1)$$

$$N = \sum_{g=0}^{L-1} n_g, \quad \text{and} \quad \sum_{g=0}^{L-1} h(g) = 1,$$

where  $n_g$  denotes the number of pixels with gray level  $g$ , whereas  $N$  represents the total number of pixels contained in the image. A mixture of Gaussian probability functions:

$$p(x) = \sum_{i=1}^K P_i \cdot p_i(x) = \sum_{i=1}^K \frac{P_i}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right) \quad (2)$$

represents a candidate histogram, with  $P_i$  as the a priori probability of class  $i$ ,  $p_i(x)$  as the probability distribution function of gray-level random variable  $x$  in class  $i$ ,  $\mu_i$  and  $\sigma_i$  as the mean and standard deviation of the  $i$ -th probability distribution function and  $K$  as the number of classes contained in the image. In addition, the constraint  $\sum_{i=1}^K P_i = 1$  must be made certain.

The Hellinger distance is used to estimate the  $3K$  ( $P_i$ ,  $\mu_i$  and  $\sigma_i$ ,  $i = 1, \dots, K$ ) parameters, comparing in such way the mixture of Gaussian functions (or candidate histogram) and the original histogram:

$$E = \sqrt{\sum_{j=1}^n \left( \sqrt{p(x_j)} - \sqrt{h(x_j)} \right)^2} \quad (3)$$

where  $p(x_j)$  is the histogram formed with the candidate Gaussian mixture and  $h(x_j)$  is the experimental histogram that corresponds to the gray level image. This is the fitness function used by the two evolutionary algorithms reported in this work.

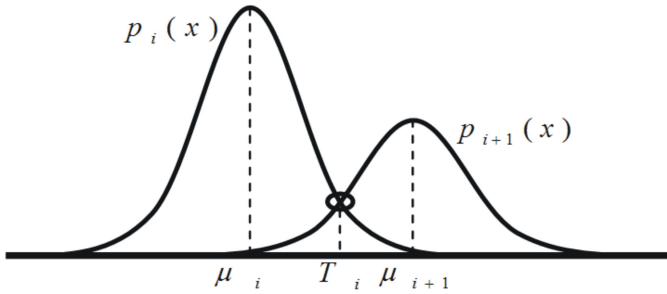
Once obtained the best histogram, next step is to determine the optimal threshold values. In what follows, will be considered that the data classes are organized such that  $\mu_1 < \mu_2 < \dots < \mu_K$ ; the threshold values can thus be calculated by estimating the overall probability error for two adjacent Gaussian functions, as follows:

$$E(T_i) = P_{i+1} \cdot E_1(T_i) + P_i \cdot E_2(T_i), \quad i = 1, \dots, K-1 \quad (4)$$

considering

$$\begin{aligned} E_1(T_i) &= \int_{-\infty}^{T_i} p_{i+1}(x) dx, \\ E_2(T_i) &= \int_{T_i}^{\infty} p_i(x) dx. \end{aligned} \quad (5)$$

$E_1(T_i)$  is the probability of mistakenly classifying the pixels in the  $(i+1)$ -th class to the  $i$ -th class, while  $E_2(T_i)$  is the probability of erroneously classifying the pixels in the  $i$ -th class to the  $(i+1)$ -th class. The  $P_j$ 's are the a-priori probabilities within the combined probability density function, and  $T_i$  is the threshold value between the  $i$ -th and the  $(i+1)$ -th classes. One  $T_i$  value is chosen such as the error  $E(T_i)$



**Fig. 1** Determination of the threshold points

is minimized. By differentiating  $E(T_i)$  with respect to  $T_i$  and equating the result to zero, it is possible to use the following equation to define the optimum threshold value  $T_i$ :

$$AT_i^2 + BT_i + C = 0 \quad (6)$$

where

$$\begin{aligned} A &= \sigma_i^2 - \sigma_{i+1}^2 \\ B &= 2 \cdot (\mu_i \sigma_{i+1}^2 - \mu_{i+1} \sigma_i^2) \\ C &= (\sigma_i \mu_{i+1})^2 - (\sigma_{i+1} \mu_i)^2 + 2 \cdot (\sigma_i \sigma_{i+1})^2 \cdot \ln \left( \frac{\sigma_{i+1} P_i}{\sigma_i P_{i+1}} \right) \end{aligned} \quad (7)$$

Even though the above quadratic equation has two possible solutions, only one of them is feasible (the positive one which falls within the interval). Figure 1 shows the determination process of threshold points.

## 2.2 Otsu's Method

As in the method aforementioned, in this also is considered a histogram  $h(g)$ , which contains  $L$  gray levels and is taken in account the total number of pixels in the image,  $N$ , calculated by  $N = h(0) + \dots + h(L-1)$ ; in this case, are supposed thresholds  $T = \{T_1, \dots, T_{m-1}\}$ , considering  $T_0 = 0$  and  $T_m = L-1$ .

A  $m$ -partition of an image  $f(x, y)$  is defined as:

$$C_i = \{g \mid g \in f(x, y), T_{i-1} < g < T_i\}, \quad i = 0, \dots, m. \quad (8)$$

The values corresponding to each element in the partition are calculated with:

$$\begin{aligned}
 q_1 &= \sum_{i=0}^{T_1} h(g_i), & \mu_1 &= \sum_{i=0}^{T_1} \frac{h(g_i) \cdot i}{q_1}, & \sigma_1^2 &= \sum_{i=0}^{T_1} \frac{(i - \mu_1)^2 \cdot h(g_i)}{q_1}, \\
 q_i &= \sum_{i=T_{i+1}}^{T_{i+1}} h(g_i), & \mu_i &= \sum_{i=T_{i+1}}^{T_{i+1}} \frac{h(g_i) \cdot i}{q_i}, & \sigma_i^2 &= \sum_{i=T_{i+1}}^{T_{i+1}} \frac{(i - \mu_i)^2 \cdot h(g_i)}{q_i},
 \end{aligned} \tag{9}$$

where  $i = 1, \dots, m - 1$ . Whether we are looking for  $K$  threshold values, we could calculate the within class variance:

$$\sigma_{WC}^2 = \sum_{j=1}^K q_j \cdot \sigma_j^2. \tag{10}$$

Also could be calculated the between class variance, although in this work the experiments are done using the within class version, and therefore working segmentation as an optimization problem, in which the next function must be minimized:

$$\sigma_{WC}^2(T_1^*, \dots, T_K^*) = \min_{0 \leq T_1 \leq \dots \leq T_K \leq L-1} \sigma_{WC}^2(T_1, \dots, T_K). \tag{11}$$

Otsu's method was proposed late in 70's by [19] as an excellent thresholding method for images, based only in the information of the histogram; nevertheless, if the number of threshold values increases, also the number of function evaluations does, due that Otsu's method is an exhaustive technique. Considering such a drawback, ABC and DE optimizing a mixture of Gaussian functions are compared with it.

### 3 Differential Evolution and Artificial Bee Colony Optimization

In this section we give the principles of operation of the two bio-inspired techniques used to find the optimal threshold to binarize an image: Differential Evolution and Artificial Bee Colony.

#### 3.1 Differential Evolution

Differential Evolution (DE) algorithm was introduced by Storn and Price in 1995 [24]. Even though it was proposed more than a decade ago, researchers' interest on this metaheuristic continues growing, due mainly to its simplicity to implement, robustness and convergence properties. This algorithm is population based, and employs mutation and crossover operations; however, the most important is the mutation, which is the central procedure [3].

The first step in DE consists on initializing a uniformly distributed random population formed by a number of parents  $N_p$ , each one with a  $D$ -dimensional vector, limited by pre specified lower ( $x\_Low_{1,j}$ ) and upper ( $x\_high_{1,j}$ ) limits:

$$\begin{aligned} x_{i,j}^k &= x\_Low_{1,j} + rand() \cdot (x\_high_{1,j} - x\_Low_{1,j}), \\ j &= 1, \dots, D, i = 1, \dots, N_p, k = 0; \end{aligned} \quad (12)$$

Generation of a mutant vector is achieved with:

$$\mathbf{v}_i^k = \mathbf{x}_{best}^k + F \cdot (\mathbf{x}_{r_1}^k - \mathbf{x}_{r_2}^k), \quad r_1, r_2 \in \{1, \dots, N_p\}, r_1 \neq r_2 \neq i, \quad (13)$$

where  $r_1$  and  $r_2$  are randomly selected integer indexes,  $\mathbf{x}_{best}^k$  represents the best population member found so far, and  $F$  is a scaling mutation factor, usually less than 1.  $N_p$  represents the number of parents. By using one or more mutant vectors and a crossover parameter, a trial vector is obtained:

$$u_{i,j}^k = \begin{cases} v_{i,j}^k & \text{if } rand() \leq Cr \text{ or } j = j_{rand} \\ x_{i,j}^k & \text{otherwise} \end{cases}, \quad (14)$$

where  $j_{rand} \in \{1, \dots, D\}$  and where the crossover constant  $Cr \in [0, 1]$  delimits the use of some parts belonging to the mutant vector that will be part of the trial vector. In the last part of the algorithm it is used a selection operator, in order to improve solutions in the minimization case, according to:

$$\mathbf{x}_i^{k+1} = \begin{cases} \mathbf{u}_i^k & \text{if } f(\mathbf{u}_i^k) < f(\mathbf{x}_i^k) \\ \mathbf{x}_i^k & \text{otherwise} \end{cases}. \quad (15)$$

In this equation,  $f$  represents the cost function; all the steps are repeated until certain criteria is reached, usually a maximum iteration number  $N_{max}$ .

### 3.2 Artificial Bee Colony Optimization

In the Artificial Bee Colony (ABC) optimization, the initial food sources are randomly initialized by the formula:

$$\begin{aligned} x_{i,j}^k &= x\_Low_{1,j} + rand() \cdot (x\_high_{1,j} - x\_Low_{1,j}), \\ j &= 1, \dots, D, i = 1, \dots, N_{fs}, k = 0; \end{aligned} \quad (16)$$

being considered that  $\mathbf{x\_high}_1$  and  $\mathbf{x\_low}_1$  are the upper and lower limits where the function to optimize is defined;  $N_{fs}$  is the number of food sources,  $D$  states for dimensions and  $k$  is the actual iteration.

Next in the algorithm, each employed bee is sent to a randomly selected food source and a neighbor is determined randomly in order to produce a modification to the source stored in its memory:

$$b_{i,j} = x_{i,j} + \varphi_{i,j} \cdot (x_{i,j} - x_{l,j}) \quad (17)$$

where  $i \in \{1, \dots, N_{eb}\}$ ,  $j \in \{1, \dots, D\}$ ,  $l$  is randomly selected,  $i \neq j$ , and  $\varphi_{i,j}$  is a random number between -1 and 1. If this modification produces a food source outside the limits, then it is set to the appropriate limit, i.e.,  $x_{high_{i,j}}$  or  $x_{low_{i,j}}$ .

Both, the source in memory and the modified one are evaluated; the bee memorizes the new position and forgets the old one. Later, employed bees return the hive and dances; and onlooker bees will choose a food source to exploit according to a probability's function:

$$q_i = \frac{fit_i}{\sum_{j=1}^{N_{fs}} fit_j}, \quad (18)$$

where  $fit_i$  represents the fitness of solution  $i$ , evaluated by the employed  $i$ , calculated by:

$$fit_i = \begin{cases} \frac{1}{f(\mathbf{x}_i)+1} & \text{if } f(\mathbf{x}_i) \geq 0 \\ 1 + abs(f(\mathbf{x}_i)) & \text{elsewhere} \end{cases}. \quad (19)$$

Later, again a neighbor is determined by the onlooker by means of Equation (17), both food sources are evaluated and the best is memorized. Finally, one scout is generated at the end of each iteration in order to explore for new food sources. The cycle is repeated until a criterion is reached.

## 4 Experimental Results

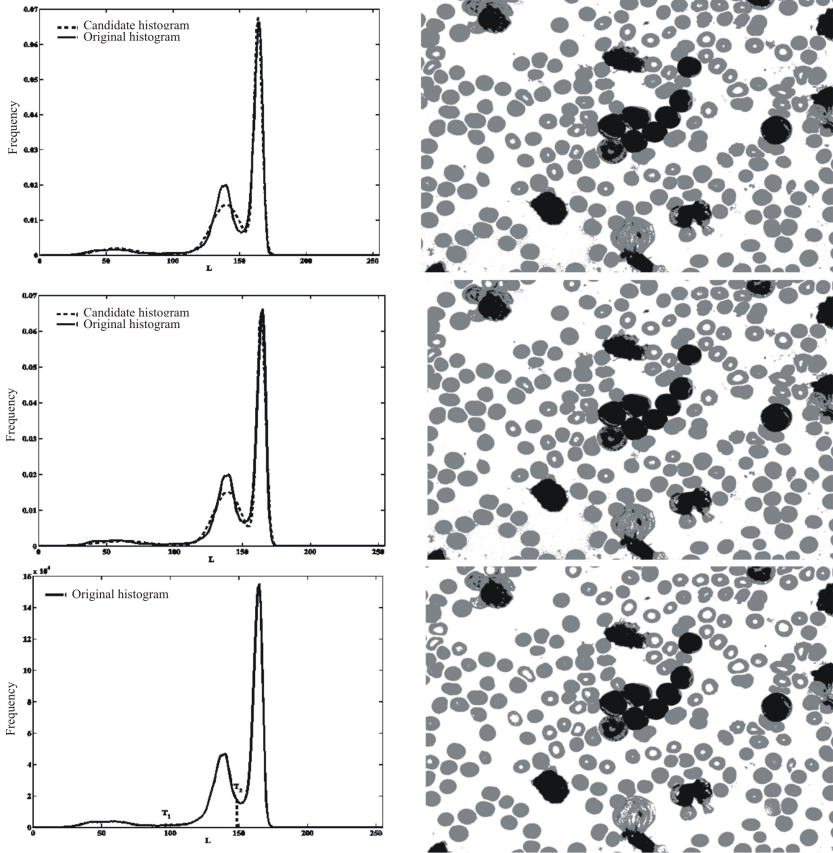
In this part several statistical results related with the number of evaluations of the objective function as well as qualitative measures of quality between segmented results and ground-truth images over a database of smear blood pictures [4] are presented. All the experiments were performed using a desktop computer with AMD Athlon II X4 2.9 GHz, with 4 GB in RAM and programmed in Matlab 7.12.0.

Figure 2 shows typical segmentation results using DE, ABC and Otsu's methods, where the main idea was to find the correct separation between two objects and the background in real images, whereas Table 1 shows particular parameters for each algorithm. Each candidate solution holds the elements

$$[P_1^i, P_2^i, P_3^i, \mu_1^i, \mu_2^i, \mu_3^i, \sigma_1^i, \sigma_2^i, \sigma_3^i]$$

and is experimentally determined a Hellinger distance of 0.1186 as stop criterion for DE and ABC.





**Fig. 2** Segmenting results corresponding, from top to bottom, to ABC, DE and Otsu

The idea behind using the same population size for all the experiments lays in the fact that in order to do the comparisons in similar circumstances for DE and ABC, the main parameter to consider is the function's evaluation number, because such parameter is considered the most expensive in the computational cost. In the first part, experiments are repeated 1000 times for each of 295 smear blood images for ALL-IDB database; we only consider such images and not all the 367 because the three classes must be more or less well defined in order to get good results with the proposal. Statistical results are shown in Table 2.

As can be seen from Table 2, DE performs around 75% less evaluations of objective functions than classic Otsu's method, compared with ABC that is around 41% better than Otsu's, considering two threshold values are searched and showing therefore that DE is computationally cheaper than ABC and Otsu's.

**Table 1** General parameters used by evolutionary algorithms

Parameter	DE,ABC: value	Observation
$N_{max}$	600	Maximum number of iterations
$x_{high_i}$	$[0.5, 0.5, 0.5, L-1, L-1, L-1, (L-1)/2, (L-1)/2, (L-1)/2]$	High limits of candidate $i$
$x_{low_i}$	$[0, 0, 0, 0, 0, 0, 0, 0, 0]$	Low limits of candidate $i$
$T$	2	Number of thresholds to find

**Table 2** Statistical results of ABC and DE over ALL-IDB database. Hereby, f-eval denotes the number of objective function evaluations

Technique	Hellinger distance		Iterations		f-eval		Execution time (s)	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
ABC	0.11859	0.000043	425	129.4	38284	11668.1	3.6175	1.0981
DE	0.11855	0.000065	176	48.5	15948	4366.0	1.2186	0.3311
OTSU	NA	NA	NA	NA	65535	NA	2.3728	0.0353

Even though the results are statistically good enough, it does not imply that quality in segmentation behaves in similar way. In such a sense, the next experimental part is related with that issue. By using a modified Hausdorff distance [6] between ground-truth and segmented images, which are obtained with DE, ABC and Otsu's techniques, the results are obtained that are shown in Table 3.

**Table 3** Quality results of ABC and DE over ALL-IDB database

Technique	Hausdorff distance	
	$\mu$	$\sigma$
ABC	42.5375	19.1111
DE	42.7095	18.7571
OTSU	38.2691	13.7078

As can be seen, differences in distances among ground-truth and segmented images are big, due mainly that ground-truth ones are obtained by human user. Some examples are shown in Figure 3. On the other hand, it is numerically seen that Otsu's result is more similar to human-segmented images and although ABC has a mean distance close to Otsu, DE has less dispersion in data; however, in quality terms, DE behavior is similar to ABC. Finally, if execution time is considered, DE outperforms ABC and Otsu's, according to results shown in Table 3. Such a time is taken since the first candidate solution is generated, until the corresponding algorithm find the best one.



**Fig. 3** White blood cell nucleus segmented examples, from left to right: Original and ground-truth images, ABC, DE and Otsu's segmentations

## 5 Conclusions and Future Work

In this paper the use of Hellinger's distance is proposed as an objective function in an evolutionary method to perform histogram-based thresholding; experimental data show that DE is less computationally expensive to solve such issue, compared with ABC and Otsu's method; however, ABC and DE behave in a similar fashion if a quality measure based in Hausdorff distance is calculated, although Otsu's method yields better results in terms of quality.

Future work involves comparing several measures of distance by using DE in order to find the most appropriate when a mixture of Gaussian functions is used to threshold images.

**Acknowledgements.** H. Sossa thanks SIP-IPN and CONACYT for the economical support under grants SIP 20111016, SIP 20121311 and CONACYT 155014 to develop the reported investigations. All authors also thank CIC-IPN and CUCEI-UDEG for the support to undertake this research.

## References

1. Arora, S., Acharya, J., Verma, A., Panigrahi, P.: Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters* 29, 119–125 (2008)
2. Bissinger, B.E., Culver, R.L., Bose, N.K.: Minimum Hellinger Distance based classification of underwater acoustic signals. In: 43rd Annual Conference on Information Sciences and Systems, CISS 2009, March 18-20, pp. 47–49 (2009)
3. Cuevas, E., Zaldívar, D., Pérez-Cisneros, M.: A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert. Syst. Appl.* 37(7), 5265–5271 (2010)

4. Labati, R., Donida Piuri, V., Scotti, F.: All-IDB: The acute lymphoblastic leukemia image database for image processing. In: 18th IEEE International Conference on Image Processing (ICIP 2011), September 11-14, pp. 2045–2048 (2011)
5. Donoho, D., Liu, R.: The automatic robustness of minimum distance functionals. *Annals of Statistics* 16, 552–586 (1988)
6. Dubuisson, M.-P., Jain, A.K.: A modified Hausdorff distance for object matching. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, October 9-13, vol. 1, pp. 566–568 (1994)
7. Halim, N.H.A., Mashor, M.Y., Abdul Nasir, A.S., Mokhtar, N.R., Rosline, H.: Nucleus segmentation technique for acute Leukemia. In: IEEE 7th International Colloquium on Signal Processing and its Applications (CSPA), March 4-6, pp. 192–197 (2011)
8. Nor Hazlyna, H., Mashor, M.Y., Mokhtar, N.R., Aimi Salihah, A.N., Hassan, R., Raof, R.A.A., Osman, M.K.: Comparison of acute leukemia Image segmentation using HSI and RGB color space. In: 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA 2010), May 10-13, pp. 749–752 (2010)
9. Janev, M., Pekar, D., Jakovljevic, N., Delic, V.: Eigenvalues Driven Gaussian Selection in continuous speech recognition using HMMs with full covariance matrices. *Applied Intelligence* 33(2), 107–116 (2010)
10. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
11. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing* 8(1), 687–697 (2008)
12. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation* 214, 108–132 (2009)
13. Kocsor, A., Tóth, L.: Application of Kernel-Based Feature Space Transformations and Learning Methods to Phoneme Classification. *Applied Intelligence* 21(2), 129–142 (2004)
14. Mezghani, N., Mitiche, A., Cheriet, M.: On-line character recognition using histograms of features and an associative memory. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 5, pp. 17–21 (2004)
15. Mohapatra, S., Patra, D.: Automated cell nucleus segmentation and acute leukemia detection in blood microscopic images. In: International Conference on Systems in Medicine and Biology (ICSMB 2010), December 16-18, pp. 49–54 (2010)
16. Mohapatra, S., Samanta, S.S., Patra, D., Satpathi, S.: Fuzzy Based Blood Image Segmentation for Automated Leukemia Detection. In: International Conference on Devices and Communications (ICDeCom 2011), February 24-25, pp. 1–5 (2011)
17. Mohapatra, S., Patra, D., Kumar, K.: Blood microscopic image segmentation using rough sets. In: International Conference on Image Information Processing (ICIIP 2011), November 3-5, pp. 1–6 (2011)
18. Olsson, R., Petersen, K., Lehn-Schioler, T.: State-Space Models: From the EM Algorithm to a Gradient Approach. *Neural Computation* 19(4), 1097–1111 (2008)
19. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics* 9(1), 62–66 (1979)
20. Park, H., Amari, S., Fukumizu, K.: Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks* 13, 755–764 (2000)
21. Park, H., Ozeki, T.: Singularity and slow Convergence of the EM algorithm for Gaussian Mixtures. *Neural Processing Letters* 29, 45–59 (2009)
22. Umebayashi, K., Lehtomaki, J., Ruotsalainen, K.: Analysis of Minimum Hellinger Distance Identification for Digital Phase Modulation. In: IEEE International Conference on Communications (ICC 2006), vol. 7, pp. 2952–2956 (2006)

23. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* 13(1), 146–168 (2004)
24. Storn, R., Price, K.: Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report (1995)
25. Schmitt, F., Priese, L.: Sky detection in CSC color images. In: VISAPP2, pp. 101–106. INSTICC Press (2009)
26. Chen, W., Fang, K.: Multilevel thresholding algorithm based on particle swarm optimization for image segmentation. In: 27th Chinese Control Conference (CCC 2008), pp. 348–351 (2008)
27. Yang, X.-S.: Review of meta-heuristics and generalized evolutionary walk algorithm. *Int. J. Bio-Inspired Comput.* 3(2) (2011)
28. Zhiwei, Y., Zhengbing, H., Huamin, W., Hongwei, C.: Automatic Threshold Selection Based on Artificial Bee Colony Algorithm. In: 3rd International Workshop on Intelligent Systems and Applications (ISA), pp. 1–4 (2011)

# Fast Circle Detection Using Harmony Search Optimization

Erik Cuevas, Humberto Sossa, Valentín Osuna, Daniel Zaldivar,  
and Marco Pérez-Cisneros

**Abstract.** Automatic circle detection in digital images has received considerable attention over the last years. Recently, several robust circle detectors, based on evolutionary algorithms (EA), have been proposed. They have demonstrated to provide better results than those based on the Hough Transform. However, since EA-detectors usually need a large number of computationally expensive fitness evaluations before a satisfying result can be obtained; their use for real time has been questioned. In this work, a new algorithm based on the Harmony Search Optimization (HSO) is proposed to reduce the number of function evaluation in the circle detection process. In order to avoid the computation of the fitness value of several circle candidates, the algorithm estimates their values by considering the fitness values from previously calculated neighboring positions. As a result, the approach can substantially reduce the number of function evaluations preserving the good search capabilities of HSO. Experimental results from several tests on synthetic and natural images with a varying complexity range have been included to validate the efficiency of the proposed technique regarding accuracy, speed and robustness.

## 1 Introduction

The problem of detecting circular features is very important for image analysis, in particular for industrial applications such as automatic inspection of manufactured products and components, aided vectorization of drawings, target detection, etc. [1]. Circular Hough transform [2] is arguably the most common technique for circle detection in digital images. A typical Hough-based approach employs an edge detector

---

Erik Cuevas · Daniel Zaldivar · Marco Pérez-Cisneros

CUCEI-UDEG, Dept. de C. Comput, Av. Revolución 1500, Guadalajara, Jal, Mexico  
e-mail: [erik.cuevas,daniel.zaldivar,marco.perez}@cucei.udg.mx](mailto:{erik.cuevas,daniel.zaldivar,marco.perez}@cucei.udg.mx)

Humberto Sossa · Valentín Osuna

CIC-IPN, Juan de Dios Batiz S/N, Col. Nva. Industrial Vallejo, Mexico City, Mexico  
e-mail: [hsossa@cic.ipn.mx](mailto:hsossa@cic.ipn.mx), [valenteos@hotmail.com](mailto:valenteos@hotmail.com)

to infer locations and radius values. Averaging, filtering and histogramming of the transformed space are subsequently applied. The approach demands a large storage space as 3-D cells to store operational parameters ( $x$ ,  $y$ ,  $r$ ), seriously constraining the overall performance to low processing speeds. In Hough Transform methods, circles parameters are poorly defined under noisy conditions [3] yielding a longer processing time which constrains their application. In order to overcome such problems, researchers have proposed new Hough transform-based (HT) approaches such as the probabilistic HT [4], the randomized HT (RHT) [5] and the fuzzy HT (FHT) [6]. In [7], Lu & Tan proposed a novel approach based on RHT called Iterative Randomized HT (IRHT) that achieves better results on complex images and noisy environments. Such implementation applies iteratively the RHT to a given region of interest which has been previously defined from the latest estimation of ellipse/circle parameters.

Alternatively to the Hough Transform, the shape recognition problem in computer vision has also been handled with EA approaches. Such methods have produced several robust circle detector using Genetic algorithms (GA) [8], Artificial Bee Colony (ABC) [9], Harmony Search (HSO) [10], Electromagnetism-Like (EMO) [11], Differential Evolution (DE) [12] and Bacterial Foraging Optimization (BFOA) [13]. They have demonstrated to give better results than those based on the Hough Transform, mainly in cluttered environments [8].

As an EA, the Harmony Search Optimization (HSO) [14] is one of the most popular methods which are based on the metaphor of the improvisation process that occurs when a musician searches for a better state of harmony. HSO generates a new candidate solution from all existing solutions. In HSO, the solution vector is analogous to the harmony in music, and the local and global search schemes are analogous to musician's improvisations. In comparison to other meta-heuristics in the literature, HSO imposes fewer mathematical requirements as it can be easily adapted for solving several sorts of engineering optimization challenges [15,16]. Furthermore, numerical comparisons have demonstrated that the evolution for the HSO is faster than GA [15,17,18], attracting ever more attention. It has been successfully applied to solve a wide range of practical optimization problems such as structural optimization, parameter estimation of the nonlinear Muskingum model, design optimization of water distribution networks, vehicle routing, combined heat and power economic dispatch, design of steel frames, bandwidth-delay-constrained least-cost multicast routing, among others.

However, one main difficulty in applying HSO (or other EA) to real-world applications is that it usually needs a large number of fitness evaluations before a satisfying result can be obtained. However, fitness evaluations are not always straightforward in many real-world applications. Either an explicit fitness function does not exist, or the evaluation of the fitness is computationally very expensive. Furthermore, since random numbers are involved in the calculation of new individuals, they may encounter the same positions (repetition) that other individuals have visited in previous iterations, especially when the individuals are confined to a small area (as in the case of images).

The problem of considering expensive fitness evaluations has already been faced in the field of evolutionary algorithms (EA) and is better known as fitness approximation [19]. In such approach, the idea is to estimate the fitness value of so many individuals as it is possible instead of evaluating the complete set. Such estimations are based on an approximate model of the fitness landscape. Thus, the individuals to be evaluated and those to be estimated are determined following some fixed criteria which depend on the specific properties of the approximate model [20]. The models involved at the estimation can be built during the actual EA run, since EA repeatedly sample the search space at different points [21]. Recently several approximation models [22-25] have been proposed to be used in combination with EA.

In this work, a new algorithm based on the Harmony Search Optimization (HSO) is proposed to reduce the number of function evaluation in the circle detection process. In order to avoid the computing the fitness value of several circle candidates, the algorithm instead of calculating them, it estimates their values by considering the fitness values from previously calculated neighboring positions. As a result, the approach can substantially reduce the number of function evaluations preserving the good search capabilities of HSO. Experimental results from several tests on synthetic and natural images with a varying complexity range have been included to validate the efficiency of the proposed technique regarding accuracy, speed and robustness.

The overall paper is organized as follows: Section 2 holds a brief description about the HSO method. In Section 3, the fitness calculation strategy for solving the expensive optimization problem is presented. The complete circle detection procedure is presented by Section 4 while Section 5 exposes the final detector algorithm as a combination of HSO and the fitness calculation strategy. Section 6 demonstrates experimental results for the proposed approach over a set of standard images and some conclusions are drawn in Section 7.

## 2 Harmony Search Optimization

In the basic HSO, each solution is called a “harmony” and is represented by an  $n$ -dimension real vector. An initial population of harmony vectors are randomly generated and stored within a Harmony Memory (HM). A new candidate harmony is thus generated from the elements in the HM by using a memory consideration operation either by a random re-initialization or a pitch adjustment operation. Finally, the HM is updated by comparing the new candidate harmony and the worst harmony vector in the HM. The worst harmony vector is replaced by the new candidate vector in case it is better than the worst harmony vector in the HM. The above process is repeated until a certain termination criterion is met. The basic HS algorithm consists of three basic phases: HM initialization, harmony vector improvisation and HM updating. The following discussion addresses details about each stage.



In the first stage, initial vector components at HM, i.e. HMS vectors, are configured. Let  $\mathbf{x}_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$  represent the  $i$ -th randomly-generated harmony vector:  $x_i(j) = l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1)$  for  $j = 1, 2, \dots, n$  and  $i = 1, 2, \dots, HMS$ , where  $\text{rand}(0, 1)$  is a uniform random number between 0 and 1. Then, the HM matrix is filled with the *HMS* harmony vectors as follows:

$$HM = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{HMS} \end{bmatrix} \quad (1)$$

In the next phase, a new harmony vector  $\mathbf{x}_{new}$  is built by applying the following three operators: memory consideration, random re-initialization and pitch adjustment. Generating a new harmony is known as 'improvisation'. In the memory consideration step, the value of the first decision variable  $x_{new}(1)$  for the new vector is chosen randomly from any of the values already existing in the current HM i.e. from the set  $\{x_1(1), x_2(1), \dots, x_{HMS}(1)\}$ . For this operation, a uniform random number  $r_1$  is generated within the range  $[0, 1]$ . If  $r_1$  is less than *HMCR*, the decision variable  $x_{new}(1)$  is generated through memory considerations; otherwise,  $x_{new}(1)$  is obtained from a random re-initialization between the search bounds  $[l(1), u(1)]$ . Values of the other decision variables  $x_{new}(2), x_{new}(3), \dots, x_{new}(n)$  are also chosen accordingly. Therefore, both operations, memory consideration and random re-initialization, can be modelled as follows:

$$x_{new}(j) = \begin{cases} x_i(j) \in \{x_1(j), x_2(j), \dots, x_{HMS}(j)\} & \text{with probability } HMCR \\ l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1) & \text{with probability } 1-HMCR \end{cases} \quad (2)$$

Every component obtained by memory consideration is further examined to determine whether it should be pitch-adjusted. For this operation, the Pitch-Adjusting Rate (*PAR*) is defined as to assign the frequency of the adjustment and the Bandwidth factor (*BW*) to control the local search around the selected elements of the HM. Hence, the pitch adjusting decision is calculated as follows:

$$x_{new}(j) = \begin{cases} x_{new}(j) = x_{new}(j) \pm \text{rand}(0, 1) \cdot BW & \text{with probability } PAR \\ x_{new}(j) & \text{with probability } 1-PAR \end{cases} \quad (3)$$

Pitch adjusting is responsible for generating new potential harmonies by slightly modifying original variable positions. Such operation can be considered similar to the mutation process in evolutionary algorithms. Therefore, the decision variable is either perturbed by a random number between 0 and *BW* or left unaltered. In order to protect the pitch adjusting operation, it is important to assure that points lying outside the feasible range  $[l, u]$  must be re-assigned i.e. truncated to the maximum or minimum value of the interval.

After a new harmony vector  $\mathbf{x}_{\text{new}}$  is generated, the harmony memory is updated by the survival of the fit competition between  $\mathbf{x}_{\text{new}}$  and the worst harmony vector  $\mathbf{x}_w$  in the HM. Therefore  $\mathbf{x}_{\text{new}}$  will replace  $\mathbf{x}_w$  and become a new member of the HM in case the fitness value of  $\mathbf{x}_{\text{new}}$  is better than the fitness value of  $\mathbf{x}_w$ .

### 3 Fitness Approximation Method

In this paper, we explore the use of a local approximation scheme based on the nearest-neighbor-interpolation (NNI) for reducing the function evaluation number. The model estimates the fitness values based on previously evaluated neighboring individuals which have been stored during the evolution process. At each generation, some individuals of the population are evaluated through the accurate (real) fitness function while the other remaining individuals are only estimated. The positions to be accurately evaluated are determined based on their proximity to the best individual or regarding their uncertain fitness value.

In a fitness approximation method, every evaluation or estimation of an individual produces a data point (individual position and fitness value) that is potentially taken into account for building the approximation model during the evolution process. Therefore, in our proposed approach, it is kept all seen-so-far evaluations in a history array  $\mathbf{T}$  which is employed to select the closest neighbor and to estimate the fitness value of a new individual.

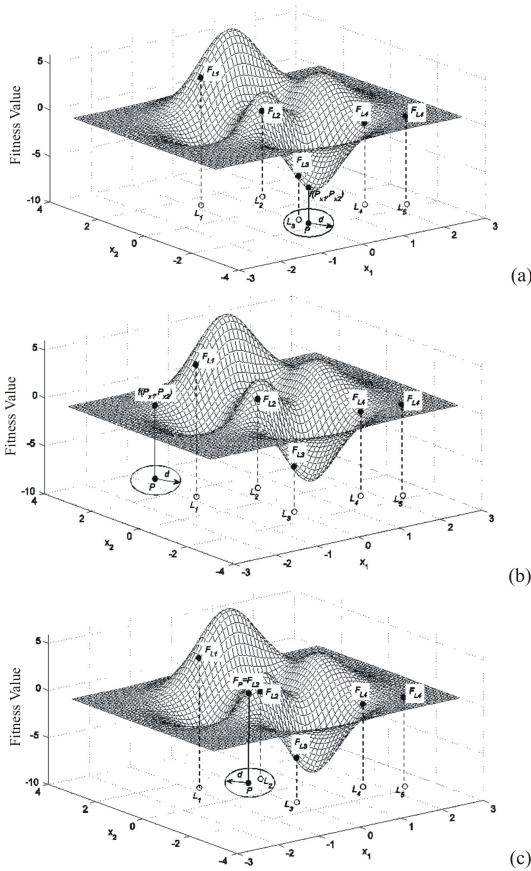
The fitness calculation model decides if an individual must be calculated or estimated based on the three following rules:

**Exploitation Rule (Evaluation).** If a new individual (search position)  $P$  is located closer than a distance  $d$  with respect to the nearest individual  $L_q$  ( $q = 1, 2, 3, \dots, m$ ; where  $m$  is the number of elements contained in  $\mathbf{T}$ ) whose fitness value  $F_{L_q}$  corresponds to the best fitness value seen so far, then the fitness value of  $P$  is evaluated using the real fitness function. Figure 1(a) draws the rule procedure.

**Exploration Rule (Evaluation).** If a new individual  $P$  is located longer than a distance  $d$  with respect to the nearest individual  $L_q$ , then its fitness value is evaluated using the real fitness function. Figure 1(b) outlines the rule procedure.

**NNI Rule (Estimation).** If a new individual  $P$  is located closer than a distance  $d$  with respect to the nearest individual  $L_q$  whose fitness value  $F_{L_q}$  does not correspond to the best fitness value, then its fitness value is estimated assigning it the same fitness that  $L_q$  ( $F_p = F_{L_q}$ ). Figure 1(c) sketches the rule procedure.

The  $d$  value controls the trade-off between the evaluation and the estimation of search locations. Typical values of  $d$  range from 1 to 4; however, in this paper, the value of 3 has been selected. The three rules show that the fitness calculation strategy is simple and straightforward. Figure 1 illustrates the procedure of fitness



**Fig. 1** The fitness calculation strategy. (a) According to the rule 1, the individual (search position)  $P$  is evaluated since it is located closer than a distance  $d$  with respect to the nearest individual location  $L_3$ . Therefore, the fitness value  $F_{L_3}$  corresponds to the best fitness value (minimum). (b) According to the rule 2, the search point  $P$  is evaluated and there is no close reference within its neighborhood. (c) According to rule 3, the fitness value of  $P$  is estimated by means of the NNI-estimator, assigning  $F_P = F_{L_2}$ .

computation for a new solution (point  $P$ ). In the problem, the objective function  $f$  is minimized with respect to two parameters ( $x_1, x_2$ ). In all figures (Figs. 1(a), (b) and (c)), the individual database array  $\mathbf{T}$  contains five different elements ( $L_1, L_2, L_3, L_4, L_5$ ) with their corresponding fitness values ( $F_{L_1}, F_{L_2}, F_{L_3}, F_{L_4}, F_{L_5}$ ). Figures 1(a) and (b) show the fitness evaluation ( $f(x_1, x_2)$ ) of the new solution  $P$ , following the rule 1 and 2 respectively, whereas Fig. 1(c) presents the fitness estimation of  $P$  using the NNI approach which is laid by rule 3.

## 4 Circle Detection Using HSO

In our approach, circles are represented by the parameters of the circle centre  $(x_0, y_0)$  and radius, considering only three non-colinear points in the edge-only space of the image. All edge pixels are stored within a vector array  $\mathbf{E} = \{e^1, e^2, \dots, e^{Np}\}$  with  $Np$  being the total number of edge pixels in the image. In turn, the algorithm stores the  $(x_v, y_v)$  coordinates of each edge pixel yielding the edge vector  $e^v$  ( $e_1^v = x_v, e_2^v = y_v$ ). In order to construct each candidate circle  $\mathbf{C}$  (or harmony within the HSO framework), indexes  $i, j$  and  $k$  representing three edge points previously stored in vector  $\mathbf{E}$  must be combined. Therefore, each harmony is encoded as one circle  $\mathbf{C} = \{e_i, e_j, e_k\}$ , which is characterized by three points, and that lie on its own circumference. Such candidate circle is labeled as a potential solution for the detection problem. Considering the configuration of the edge points in Fig. 2, the circle centre  $(x_0, y_0)$  and the radius  $r$  of  $\mathbf{C}$  can be calculated using simple geometric equations [37].

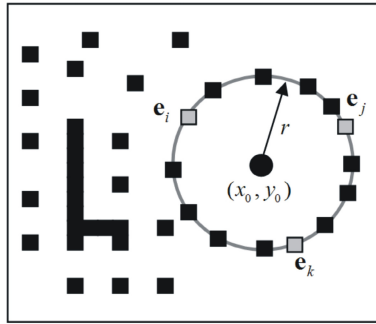


Fig. 2 Circle candidate formed from the combination of points  $e_i, e_j$ , and  $e_k$

### 4.1 Objective Function

A circumference may be calculated as a virtual shape as a way to measure the matching factor between the candidate circle  $\mathbf{C}$  and the actual circle within the image, i.e. it must be validated if such circular shape actually exists within the edge-only image. The test for such points is verified using  $\mathbf{S} = \{s^1, s^2, \dots, s^{Ns}\}$ , where  $Ns$  represents the number of testing points.

The set  $\mathbf{S}$  is generated by means of Midpoint Circle Algorithm (MCA) [27] which calculates the vector  $\mathbf{S}$  representing a circle from parameters  $(x_0, y_0)$  and  $r$ . The matching function, also known as the objective function  $J(\mathbf{C})$ , represents the error resulting from pixels  $\mathbf{S}$  of a given circle candidate and  $\mathbf{C}$  including only pixels that really exist in the edge image, yielding:

$$J(\mathbf{C}) = 1 - \frac{\sum_{v=1}^{Ns} V(s^v)}{Ns} \quad (4)$$

where  $V(s^v)$  is a function that verifies the pixel existence in  $s^v$ , such as:

$$V(s^v) = \begin{cases} 1 & \text{if the pixel in position } (x_v, y_v) \text{ exists} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The algorithm seeks to minimize  $J(\mathbf{C})$ , since a smaller value implies a better response (minimum error) of the ‘‘circularity’’ operator. The optimization process can thus be stopped either because the maximum number of epochs is reached or because the best individual is found.

## 5 Final Detection Algorithm

The goal of our HSO-detector is to reduce the number of evaluations of the objective function avoiding any performance loss and achieving an accurate solution. The HSO-detector is listed below:

Step 1: Set the HSO parameters. Initializing the harmony memory with  $HMS$  individuals where each decision variable  $p_i, p_j$ , and  $p_k$  of the candidate circle  $\mathbf{C}_a$  is set randomly within the interval  $[1, E_p]$ . All values must be integers. Considering  $a = (1, 2, \dots, HMS)$ .

Step 2: Evaluate or estimate (according to the fitness calculation strategy, see Section 3) the objective value  $J(\mathbf{C}_a)$  for all  $HMS$  individuals and determine the candidate solution  $\mathbf{C}_w$  of  $HMS$  holding the worst objective value. Update the new evaluations in the individual database array  $\mathbf{T}$ .

Step 3: Improve a new harmony  $\mathbf{C}_{new}$  such that:

for ( $j = 1$  to 3) do

if ( $r_1 < HMCR$ ) then  $\mathbf{C}_{new}(j) = \mathbf{C}_a(j)$  where  $a \in (1, 2, \dots, HMS)$

if ( $r_2 < PAR$ ) then  $\mathbf{C}_{new}(j) = \mathbf{C}_{new}(j) \pm r_3 \cdot BW$  where  $r_1, r_2, r_3 \in (0, 1)$

if  $\mathbf{C}_{new}(j) < l(j)$

$\mathbf{C}_{new}(j) = l(j)$

end if

if  $\mathbf{C}_{new}(j) > u(j)$

$\mathbf{C}_{new}(j) = u(j)$

end if

Update  $BW$  according to Eq. (4)

end if

else

$\mathbf{C}_{new}(j) = 1 + \text{round}(r \cdot E_p)$ , where  $r \in (0, 1)$

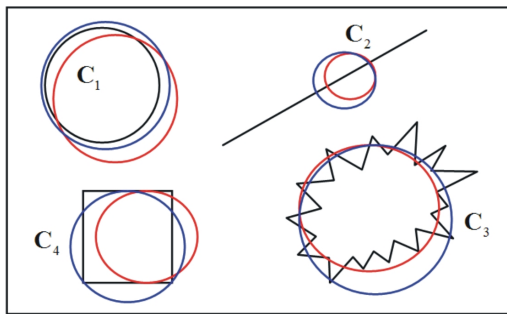
end if

end for

Step 4: Update  $HM$ . In case that the fitness value (evaluated or approximated, according to the fitness calculation strategy, see Section 3) of the new solution  $C_{new}$ , is better than the solution  $C_w$ , such position is selected as an element of  $HM$ , otherwise the solution  $C_w$  remains. Update the new evaluations in the individual database array  $T$ .

Step 5: If  $NI$  is completed then return the best harmony vector  $C_b$  in the  $HM$  (a circle contained in the image); otherwise go back to Step 2.

The incorporated fitness calculation strategy, seen from an optimization perspective, favors the exploitation and exploration in the search process. For the exploration, the method evaluates the fitness function of new circle candidates which have been located far away from previously calculated positions. Additionally, it also estimates those which are closer. For the exploitation, the proposed method evaluates the actual fitness function of those new circle candidates which are located nearby the position that holds the minimum fitness value seen-so-far, aiming to improve its minimum. Therefore, the proposed fitness calculation scheme allows estimating most of fitness values reducing the computational overhead at each generation. Figure 3 presents the procedure of fitness computation for different new circle candidates. In the Figure, the black shapes represent the original image, blue circles the solutions already stored in the array  $T$  meanwhile the red circles the new circles candidates. According to the proposed schema only the circle  $C_1$  will be calculated, while the circles  $C_2$ ,  $C_3$  and  $C_4$  will be estimated (saving important computational time).



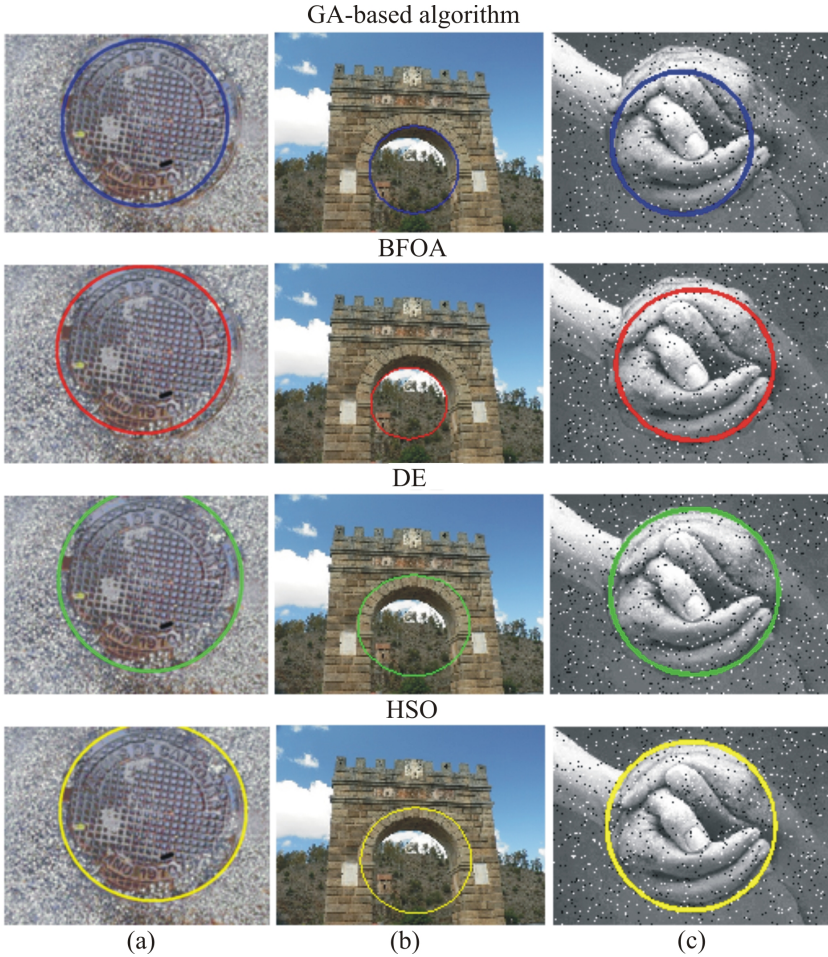
**Fig. 3** The fitness calculation strategy in circle detection

## 6 Experimental Results

Table 1 summarizes the  $HSO$  parameter set that are considered in this work and kept for all test images through all experiments.

**Table 1** Parameter setup for the HSO detector

HMS	HMCR	PAR	BW	NI
100	0.7	0.3	5	200

**Fig. 4** Experimental image set

This section presents a comparison of the proposed approach to some other algorithms such as the BFAO detector [13], the GA-based algorithm [8] and the DE method [12] over a challenging image set. For the BFAO detector and the GA-based algorithm, the parameter values are configured as suggested in their respective references. Figure 4 presents a visual comparison among the detectors, considering

three natural images (a)-(c). In order to test the robustness of our algorithm, salt and pepper noise has been added to the last image (Fig. 4c) prior to the experiment.

Since the present approach aims to accelerate the HAS-detector, the execution time is considered as the main performance index. However, considering that the execution time largely depends on how optimal each algorithm's code is implemented, the number of function evaluation ( $FE$ ) is rather employed as a temporal criterion. Therefore,  $FE$  reports the number of evaluations that the algorithm executes until the number of iterations ( $NI$ ) has been reached (200).

Table 2 presents the number of function evaluations ( $FE$ ). Such numbers represent the averaged values obtained through 35 different executions for each algorithm. As it can be seen, the number of function evaluations for the proposed algorithm is of 52 evaluations. In contrast, the other algorithms maintain a bigger number.

**Table 2** The averaged function evaluations for the GA-based algorithm, the BFOA method, the DE approach and the proposed HSO algorithm, considering nine test images shown by Figure 3

Image	function evaluations ( $FE$ ) $\pm$ Standard deviation			
	GA	BFOA	DE	HSO
(a)	40000 $\pm$ (0)	25000 $\pm$ (0)	20000 $\pm$ (0)	1200 $\pm$ (59)
(b)	40000 $\pm$ (0)	25000 $\pm$ (0)	20000 $\pm$ (0)	986 $\pm$ (95)
(c)	40000 $\pm$ (0)	25000 $\pm$ (0)	20000 $\pm$ (0)	1050 $\pm$ (78)

## 7 Conclusions

In this work, a new algorithm based on the Harmony Search Optimization (HSO) is proposed to reduce the number of function evaluation in the circle detection process. In order to avoid the computing the fitness value of several circle candidates, the algorithm instead of calculating them, it estimates their values by considering the fitness values from previously calculated neighboring positions. As a result, the approach can substantially reduce the number of function evaluations preserving the good search capabilities of HSO. Experimental results from several images with a varying complexity range have been included to validate the efficiency of the proposed technique regarding accuracy, speed and robustness.

**Acknowledgements.** H. Sossa thanks SIP-IPN and CONACYT for the economical support under grants SIP 20111016, SIP 20121311 and CONACYT 155014 to develop the reported investigations. All authors also thank CUCEI-UDEG and CIC-IPN for the support to undertake this research.



## References

1. da Fontoura Costa, L., Marcondes Cesar Jr., R.: *Shape Analysis and Classification*. CRC Press, Boca Raton (2001)
2. Muammar, H., Nixon, M.: Approaches to extending the Hough transform. In: Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP, vol. 3, pp. 1556–1559 (1989)
3. Atherton, T.J., Kerbyson, D.J.: Using phase to represent radius in the coherent circle Hough transform. In: Proc. IEE Colloquium on the Hough Transform. IEEEE, London (1993)
4. Shaked, D., Yaron, O., Kiryati, N.: Deriving stopping rules for the probabilistic Hough transform by sequential analysis. *Comput. Vision Image Understanding* 63, 512–526 (1996)
5. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Lett.* 11(5), 331–338 (1990)
6. Han, J.H., Koczy, L.T., Poston, T.: Fuzzy Hough transform. In: Proc. 2nd Int. Conf. on Fuzzy Systems, vol. 2, pp. 803–808 (1993)
7. Lu, W., Tan, J.L.: Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT). *Pattern Recognition* 41(4), 1268–1279 (2008)
8. Ayala-Ramirez, V., Garcia-Capulin, C.H., Perez-Garcia, A., Sanchez-Yanez, R.E.: Circle detection on images using genetic algorithms. *Pattern Recognition Letters* 27, 652–657 (2006)
9. Cuevas, E., Sencin-Echauri, F., Zaldivar, D.: Pérez-Cisneros, M.: Multi-circle detection on images using artificial bee colony (ABC) optimization. *Soft Comput.* 16(2), 281–296 (2012)
10. Cuevas, E., Ortega-Sánchez, N., Zaldivar, D., Pérez-Cisneros, M.: Circle Detection by Harmony Search Optimization. *Journal of Intelligent & Robotic Systems* 66, 359–376 (2012)
11. Cuevas, E., Oliva, D., Zaldivar, D., Pérez-Cisneros, M., Sossa, H.: Circle detection using electro-magnetism optimization. *Information Sciences* 182(1), 40–55 (2012)
12. Cuevas, E., Zaldivar, D., Pérez-Cisneros, M., Ramirez-Ortegon, M.: Circle detection using discrete differential evolution optimization. *Pattern Anal. Appl.* 14(1), 93–107 (2011)
13. Dasgupta, S., Das, S., Biswas, A., Abraham, A.: Automatic circle detection on digital images whit an adaptive bacterial foraging algorithm. *Soft Computing* (2009), doi:10.1007/s00500-009-0508-z
14. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *Simulations* 76, 60–68 (2001)
15. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* 188, 1567–1579 (2007)
16. Omran, M.G.H., Mahdavi, M.: Global-best harmony search. *Appl. Math. Comput.* 198, 643–656 (2008)
17. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization, harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* 194, 3902–3933 (2005)
18. Lee, K.S., Geem, Z.W., Lee, S.H., Bae, K.W.: The harmony search heuristic algorithm for discrete structural optimization. *Eng. Optim.* 37, 663–684 (2005)
19. Jin, Y.: Comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 9, 3–12 (2005)
20. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 61–70 (2011)

21. Branke, J., Schmidt, C.: Faster convergence by means of fitness estimation. *Soft Computing* 9, 13–20 (2005)
22. Zhou, Z., Ong, Y., Nguyen, M., Lim, D.: A Study on Polynomial Regression and Gaussian Process Global Surrogate Model in Hierarchical Surrogate-Assisted Evolutionary Algorithm. In: *IEEE Congress on Evolutionary Computation (ECiDUE 2005)*, Edinburgh, United Kingdom, September 2-5 (2005)
23. Ratle, A.: Kriging as a surrogate fitness landscape in evolutionary optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 15, 37–49 (2001)
24. Lim, D., Jin, Y., Ong, Y., Sendhoff, B.: Generalizing Surrogate-assisted Evolutionary Computation. *IEEE Transactions on Evolutionary Computation* 14(3), 329–355 (2010)
25. Ong, Y., Lum, K., Nair, P.: Evolutionary Algorithm with Hermite Radial Basis Function Interpolants for Computationally Expensive Adjoint Solvers. *Computational Optimization and Applications* 39(1), 97–119 (2008)
26. Luoa, C., Shao-Liang, Z., Wanga, C., Jiang, Z.: A metamodel-assisted evolutionary algorithm for expensive optimization. *Journal of Computational and Applied Mathematics* (2011), doi:10.1016/j.cam.2011.05.047
27. Bresenham, J.E.: A Linear Algorithm for Incremental Digital Display of Circular Arcs. *Communications of the ACM* 20, 100–106 (1977)
28. Van-Aken, J.R.: An Efficient Ellipse Drawing Algorithm. *CG&A* 4, 24–35 (1984)

# Particle Swarm Optimization Applied to Interferogram Demodulation

Julio Jiménez, Humberto Sossa, and Francisco Cuevas

**Abstract.** A particle swarm optimization (PSO) based method to carry out fringe pattern demodulation is described. A particle swarm is codified with the parameters of the function that estimates the phase. A fitness function is established to evaluate the particles, which considers: (a) the closeness between the observed fringes and the recovered fringes, (b) the phase smoothness, (c) the prior knowledge of the object as its shape and size. The swarm of particles evolves until a fitness average threshold is obtained. We demonstrate that the method is able to successfully demodulate noisy fringe patterns and even a one-image closed-fringe pattern.

## 1 Introduction

It is known that in optical metrology a fringe pattern can be represented using the total irradiance, using the following mathematical expression:

$$I(x,y) = a(x,y) + b(x,y)\cos[\phi(x,y)], \quad (1)$$

where  $(x,y)$  are integer values representing the coordinates of the pixel location in the fringe image,  $a(x,y)$  is the background illumination,  $b(x,y)$  is the amplitude modulation (e.g., this factor is related with the surface reflectance) and  $\phi(x,y)$  is the phase term related to the physical quantity being measured.

---

Julio Jiménez

CIC-IPN, Juan de Dios Batiz S/N, Col. Nva. Industrial Vallejo, Mexico City, Mexico  
And CIO. León, Guanajuato, Mexico, 37150, Mexico

e-mail: [jvielma@cio.mx](mailto:jvielma@cio.mx)

Humberto Sossa

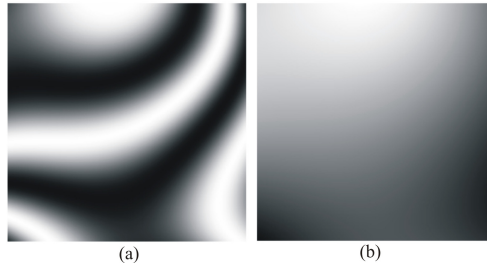
CIC-IPN, Juan de Dios Batiz S/N, Col. Nva. Industrial Vallejo, Mexico City, Mexico

e-mail: [hsossa@cic.ipn.mx](mailto:hsossa@cic.ipn.mx)

Francisco Cuevas

CIO. León, Guanajuato, Mexico, 37150, Mexico

e-mail: [fjcuevas@cio.mx](mailto:fjcuevas@cio.mx)



**Fig. 1** (a) Fringe pattern, and (b) its phase map

The purpose of any interferometric technique is to determine the phase term, which is related to the physical quantity, being measured. Fig. 1(a) shows an interferogram and its associated phase  $\phi(x,y)$  in Fig. 1(b). One way to calculate the phase term  $\phi(x,y)$  is by means of the phase-shifting technique (PST) as described in [1]-[4]. A drawback is that at least three phase-shifted interferograms are needed. The phase shift among interferograms must be known and experimentally controlled. This technique can be used when mechanical conditions are met throughout the interferometric experiment.

On the other hand, when the above mentioned conditions are not covered, other techniques can be used to estimate the phase term from a single fringe pattern. In [5] and [6] and [7], authors use Fourier transform, while in [8], phase locked loop technique is adopted. However, these techniques work well only if the analyzed interferogram has a carrier frequency, a narrow bandwidth and the signal has low noise. Moreover, these methods fail for phase calculation in a closed-fringe pattern. Additionally, the Fourier and synchronous methods estimate the wrapped phase due to the arctangent function used during the phase calculation, so an additional unwrapping procedure is required [9]. The unwrapping process is difficult when the fringe pattern includes high amplitude noise, which causes differences greater than  $2\pi$  radians between adjacent pixels [10], [11] and [12]. In the PLL technique, the phase is estimated by following the phase changes of the input signal by varying the phase of a computer simulated oscillator (VCO), such that the phase error between the fringe pattern and VCOs signal vanishes.

Recent techniques make use of soft computing algorithms like neural networks and genetic algorithms (GA). In the neural network technique [13] and [14], a multi-layer neural network (MLNN) is trained by using fringe patterns and the phase gradients associated with them, from calibrated objects. After the training, the MLNN can estimate the phase gradient when the fringe pattern is presented in the MLNN input.

GA based methods such as a single interferogram in [15] and [16], with window fringe pattern demodulating (WFPD) technique in [17] and [18], approximate the phase through the estimation of parametric functions. The functions can be Bessel in case of fringes coming from a vibrating plate experiment, or Zernike polynomials,

in an optical testing experiment. In the case when not much information is known about the experiment, a set of low degree polynomials  $p(a, x, y)$  can be used. A population of chromosomes is codified with the function parameters that estimate the phase. A fitness function is established to evaluate the chromosomes, which considers the same aspects as the cost function in a regularization technique. The population of chromosomes evolves until a fitness average threshold is obtained. The method can demodulate noisy, closed fringe patterns and so no further unwrapping is needed.

In this paper we present a variation of the WFPD method introduced by Cuevas et al. in [18]. The new proposal is applied to demodulate complex fringe pattern using a differential evolution technique to fit a polynomial; it also allows creating an automatic fringe counting based on digital image processing. In addition we use low resolution versions of the interferogram for the recovery of the phase, in other words we use sub-sampled images. Results using closed and under-sampled computer generated fringe patterns are presented.

## 2 Particle Swarm Optimization

Particle swarm optimization was proposed by Kennedy and Eberhart in [19] and [20] for solving optimization problems. Each particle moves in the search space with a velocity according to its own previous best solution and its groups previous best solution. The dimension of the search space can be any positive integer. Following Eberhart and Kennedys naming conventions  $D$  is the dimension of the search space. The  $i^{th}$  particle is represented as  $A_i = (a_{i1}, a_{i2}, \dots, a_{iD})$  and the best particle of the swarm, i.e. the particle with the lowest function value, is denoted by index  $g$ . The best previous position (i.e. the position corresponding to the best function value) of the  $i^{th}$  particle is recorded and represented as  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ , and the position change (velocity) of the  $i^{th}$  particle is  $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ . Each particle updates its position with the following two equations:

$$v_{id}(t+1) = \omega a_{id} + c_1 \phi_1 (p_{id} - a_{id}(t)) + c_2 \phi_2 (p_{gd} - a_{id}(t)) \quad (2)$$

$$a_{id}(t+1) = a_{id}(t) + v_{id}(t+1) \quad (3)$$

where for each particle  $i$  is  $a_i$  the position,  $v_{id}$  the velocity,  $p_{id}$  the best position of a particle,  $p_{gd}$  the best position within the swarm,  $c_1$  and  $c_2$  are positive constants containing the balance factors between the effect of self-knowledge and social knowledge in moving the particle towards the target.  $\phi_1$  and  $\phi_2$  are random numbers between 0 and 1, and  $\omega$  is inertia weight. Within the update of the particles, the velocity is denoted as the momentum which the force is pulling the particle to continue its current direction. The best position of a particle is cognitive component and this force emerges from the tendency to return to its own best solution found so far and the best position of a swarm is the social component. This is the force emerging from the attraction of the best solution found so far in its neighborhood.

### 3 PSO Applied to Phase Recovery

The purpose in any application of PSO is to evolve a particle swarm of size  $P$  (which codifies  $P$  possible solutions to the problem) using update velocity and position of each particle, with the goal of optimizing a fitness function that resolves the problem. In phase demodulation from fringe patterns, the phase data can be approximated by the selection from one of several fitting functions. The fitness function is modeled by the following considerations: a) The similarity between the original fringe image and the genetic generated fringe image, and b) the smoothness in the first and second derivatives of the fitting function. The fitness function used in this investigation is as follows:

$$U(a^p) = \max_p \left( \sum_{y=1}^{R-1} \sum_{x=1}^{C-1} ((I_N(x,y) - \cos(f(a^p, x, y)))^2 + \lambda [(f(a^p, x, y) - f(a^p, x-1, y))^2 + (f(a^p, x, y) - f(a^p, x, y-1))^2]) \right) m(x, y). \quad (4)$$

The first term inside the first parenthesis compares the RMS error between the original fringe pattern and the fringe pattern obtained from the estimated phase. This term attempts to keep the local fringe model close to the observed irradiances in least-squares sense, while the second term is a local discrete difference, which enforces the assumption of smoothness and continuity of the detected phase. The term  $\lambda$  is a smoothness weight factor (it should be clear for the reader that a higher value of parameter  $\lambda$  implies a smoother function to be fitted).

As it was said earlier, PSO is used to find the function parameters, in this case, vector  $a$ . If we use this function, the particle can be represented as:

$$a = [a_0, a_1, \dots, a_q] \quad (5)$$

A  $k$ -bit long bit-string is used to codify a particle value; then, the particle has  $q \times k$  bits in length. We define the search space for these parameters. The bit-string codifies a range within the limits of each parameter. The decoded value of the  $i^{\text{th}}$  parameter will use the methodology introduced by Toledo and Cuevas in [18] is:

$$a_i = L_i^B + \frac{L_i^U - L_i^B}{2^k - 1} N_i \quad (6)$$

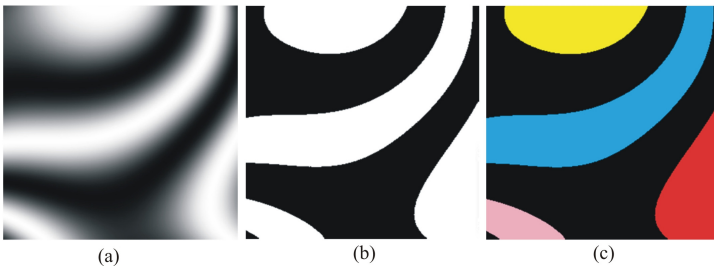
where  $a_i$  is the  $i^{\text{th}}$  parameter real value,  $L_i^B$  is the  $i^{\text{th}}$  bottom limit,  $L_i^U$  is the  $i^{\text{th}}$  upper limit, and  $N_i$  is the decimal basis value. These maximum values can be expressed as:

$$L_0^B = -\pi, \quad L_0^U = \pi, \quad L_i^U = -L_i^B, \quad L_i^U = \frac{4\pi F}{R1_i^m C1_i^n} \quad (7)$$

where  $F$  is twice the maximum number of fringes on the window, the equation is expressed in [17]:

$$F = 2 \times \max(F_x, F_y, \sqrt{F_x^2 + F_y^2}) \quad (8)$$

Finding automatically the value for  $F$  is not an easy problem to solve. To our knowledge no single method exists. To get the maximum number of fringes in an image, we propose to combine image thresholding and connected component labeling as described in [21], [22] and [23]. An example of the application of Otsus procedure onto an image is shown in Fig. 2(b). The corresponding connected component labeled image is shown in Fig. 2(c).



**Fig. 2** a) Image of fringe pattern, b) binary image using the Otsus method, c) labeling image with the result of 4 fringes in the image

## 4 PSOs Convergence

PSOs convergence depends mainly on swarm size. Large swarm convergence in a smaller number of iterations, but the processing time is increased. To stop the PSO process, different convergence measures can be employed. In this paper we have used a relative comparison between the fitness function value of the best vectors in the population and value  $\alpha$ , as follows:

$$\left| \frac{\alpha - U(a^*)}{\alpha} \right| < \varepsilon \quad (9)$$

where  $U(a^*)$  is the fitness function value of the best vectors in the population in the current iteration, and  $\varepsilon$  is the relative error tolerance. Additionally, we can stop the process in a specified number of iterations if equation (9) is not satisfied.

## 5 Experimental Results

The proposed method was applied to estimate the phase for a closed fringe pattern. We used a particles swarm size of 100, inertia was chosen in the range [0.1 to 0.9],

and velocity a number in the range [0.0001 to 0.0009]. In each particle, the coded coefficients of a fourth degree polynomial were included. The following polynomial was coded in each particle:

$$p_4(x,y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 + a_9y^3 + a_{10}x^4 + a_{11}x^3y + a_{12}x^2y^2 + a_{13}xy^3 + a_{14}y^4 \quad (10)$$

The 15 coefficients were configured in each particle inside the swarm to be evolved. As real interferograms present low contrast, and to show that our proposal performs efficiently, a low noisy closed fringe pattern was generated using the following expression:

$$I(x,y) = 127 + 63\cos(P_4(x,y) + \eta(x,y)) \quad (11)$$

where

$$p_4(x,y) = 0 - 0.7316x - 0.2801y + 0.0065x^2 - 0.00036xy - 0.0372y^2 + 0.00212x^3 + 0.000272x^2y + 0.001xy^2 - 0.002y^3 + 0.000012x^4 + 0.00015x^3y + 0.00023x^2y^2 + 0.00011xy^3 + 0.000086y^4 \quad (12)$$

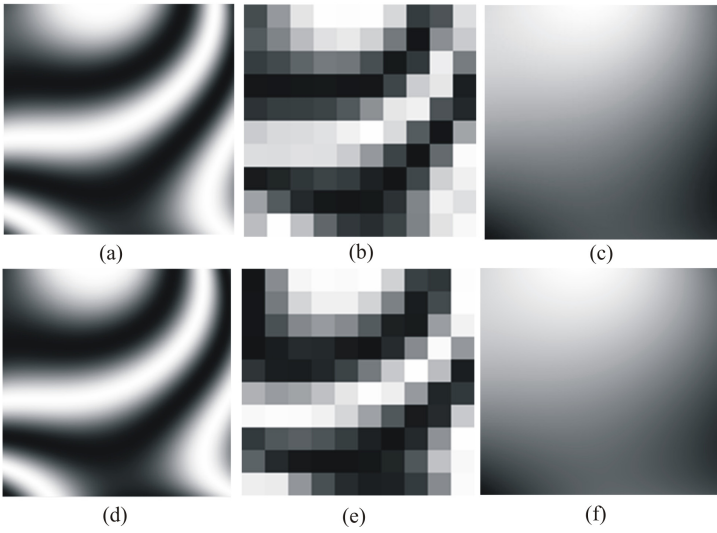
Term  $\eta(x,y)$  is the uniform additive noise in the range  $[-2\text{radians to } 2\text{radians}]$ . Additionally, the fringe pattern was generated with a low resolution of 8 pixels. In this case, we used a parameter search range of  $[-1 \text{ to } 1]$ . The swarm of particles evolved until the number of iterations and relative error tolerance  $\varepsilon$  was 0.05 in equation (9).

The fringe pattern and the phase field of the computer generated interferogram are shown in Figs. 3(a) and 3(b), respectively. The PSO technique was used to recover the phase from the fringe pattern. The fringe pattern and the phase estimated by PSO are shown in Figures 3(d), (e) and 3(f).

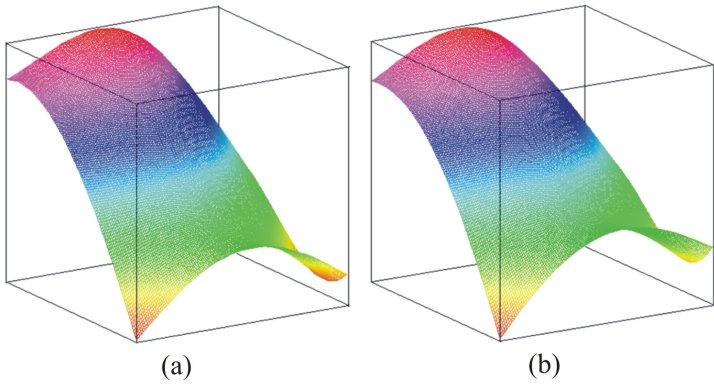
The phase map observed in 3D is shown in Fig. 4(a), and phase map in 3D estimated by PSO in Fig. 4(b). The normalized RMS error was 0.12 radians and the peak-to-valley error was 0.94 radians.

Additionally, our method was compared with that proposed by Toledo and Cuevas in [17], which is based on genetic algorithms. We took into consideration the settings of GA parameters used in [17]. Eight parameters were adjusted: number of generations, number of population, cross and mutation rate, type of selection, mutation rate and type of cross. In our case, only four parameters were adjusted: iterations (generations), swarm (population), inertia and speed. Finally, during the test an error of 0.4281 was obtained with the GA-based method. With our PSO based proposal we obtained an error of 0.313.





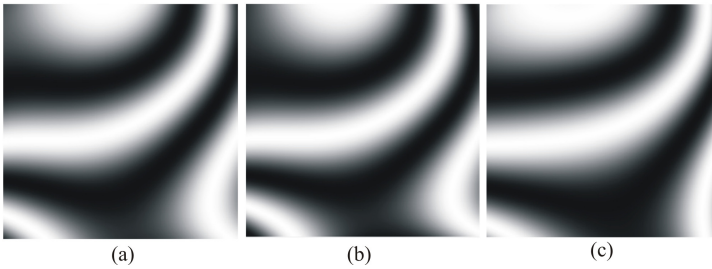
**Fig. 3** (a) Observed fringe pattern, (b) Observed fringe pattern in low resolution, (c) its phase map. (d) Estimated fringe pattern by PSO, (e) in low resolution and (g) its phase map



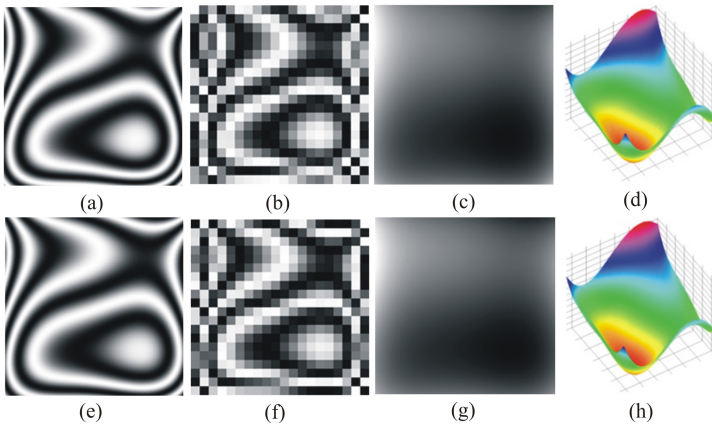
**Fig. 4** (a) Phase map from observed image, (b) Estimated phase map by PSO

Fig. 5(a) shows the original interferogram. Figs. 5(b) and 5(c) illustrate the result obtained by our method and the result obtained with the GA method introduced in [18].

The interferogram demodulation in comparison was almost identical, but the big difference is the image input was because the technique with PSO was recovered with an image sub-sampled low level, that has a serious problem of sub-Nyquits in that it no longer distinguishes fringes.



**Fig. 5** (a) Observed fringe pattern, (b) estimated fringe pattern by PSO, (c) estimated fringe pattern by GA



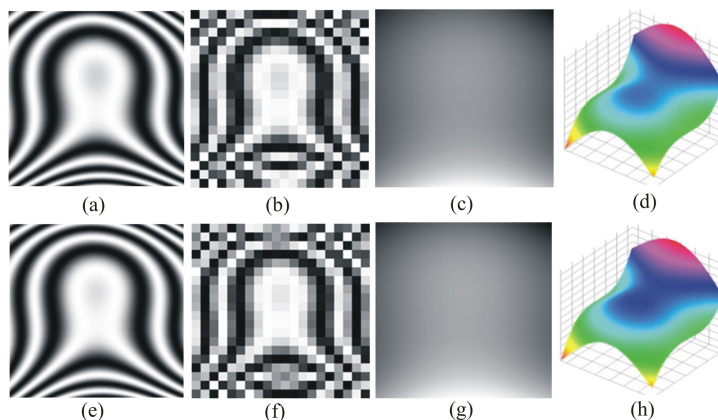
**Fig. 6** (a) Observed fringe pattern, (b) observed fringe pattern in low resolution, (c) its phase, (d) phase in 3D, (e) estimated fringe pattern by PSO, (f) in low resolution, (g) its phase map, and (h) phase in 3D

The proposed methodology was applied to other images to show its performance. For this, refer to Figs. 6 and 7.

The use of a sub-sample with high problem of sub-Nyquits is something where traditional techniques (Fourier method, Synchronous method and the phase locked loop method) fail, and instead the techniques that use GA's have a sub-sampling Nyquits above the limit (one fringe per pixel).

Compared with other literature methods, our method has the advantage that, using a single image, it does not apply any module of unwrapping to the phase and that the polynomial is directly the phase term, it can work with images with high sub-Nyquits, a problem that traditional methods have so far failed to solve.

The execution time is considered fast compared to methods using GAs which is due to the encoding and image size.



**Fig. 7** (a) Observed fringe pattern, (b) observed fringe pattern in low resolution, (c) its phase, (d) phase in 3D, (e) estimated fringe pattern by PSO, (f) in low resolution, (g) its phase map, and (h) phase in 3D

## 6 Conclusions

A PSO based technique was applied to recover the modulating phase from closed and noisy fringe patterns. A fitness function, which considers the prior knowledge of the object being tested, is established to approximate the phase data. In this work a fourth degree polynomial was used to fit the phase.

A swarm of particles was generated to carry out the optimization process. Each particle was formed by a codified string of polynomial coefficients. Then, the swarm of particles evolved used velocity, position and inertial.

The proposal works successfully where other techniques fail (Synchronous and Fourier methods). This is the case when a noisy, wide bandwidth and/or closed fringe pattern is demodulated. Regularization techniques can be used in these cases but the proposal has the advantage that the cost function does not depend upon the existence of derivatives and restrictive requirements of continuity (gradient descent methods).

Since PSO works with a swarm of possible solutions instead of a single solution, it avoids falling in a local optimum. Additionally, no filters and no thresholding operators were required, in contrast with the fringe-follower regularized phase tracker technique.

PSO has the advantage that if the user has prior knowledge of the object shape, then a better suited fitting parametric function can be used instead of a general polynomial function. Additionally, due to the fact that the PSO technique gets the parameters of the fitting function, it can be used to interpolate sub-pixel values and to increase the original phase resolution or interpolate where fringes do not exist or are not valid. A drawback is the selection of the optimal initial PSO parameters (such as swarm size, inertial, velocity) that can increase the convergence speed.

**Acknowledgements.** H. Sossa thanks SIP-IPN and CONACYT for the economical support under grants SIP 20111016, SIP 20121311 and CONACYT 155014 to develop the reported investigations. We wish to thank the Centro de Investigación en Computación of the I.P.N. and the Centro de Investigaciones en Óptica on supporting us on this major accomplishment. Finally, thanks to the anonymous reviewers who have given us their constructive criticism on the improvement of this assignment.

## References

1. Malacara, D., Servin, M., Malacara, Z.: *Interferogram Analysis for Optical Testing*. Marcel Dekker (ed.). CRC Press, New York (1998)
2. Robinson, D.W., Reid, G.T.: *Interferogram Analysis: Digital Fringe Measurement Techniques*. IOP Publishing, London (1993)
3. Creath, K.: Phase measurement interferometry techniques. In: Wolf, E. (ed.) *Progress in Optics*, vol. 26, pp. 348–393. Elsevier, Amsterdam (1988)
4. Creath, K.: *Interferogram Analysis*. Robinson, D., Reid, G.T. (eds.), p. 94. IOP Publishing, London (1993)
5. Takeda, M., Ina, H., Kobayashi, S.: Fourier-transform method of fringe-pattern analysis for computer based topography and interferometry. *J. Opt. Soc. Am.* 72, 156–160 (1981)
6. Su, X., Chen, W.: Fourier transform profilometry: a review. *Opt. Laser Eng.* 35, 263–284 (2001)
7. Womack, K.H.: Interferometric phase measurement using spatial synchronous detection. *Opt. Eng.* 23, 391–395 (1984)
8. Servin, M., Rodriguez-Vera, R.: Two dimensional phaselocked loop demodulation of interferogram. *J. Mod. Opt.* 40, 2087–2094 (1993)
9. Ghiglia, D.C., Pritt, M.D.: *Two-dimensional Phase Unwrapping*. John Wiley & Sons, Inc., New York (1998)
10. Su, X., Xue, L.: Phase unwrapping algorithm based on fringe frequency analysis in Fourier-transform profilometry. *Opt. Eng.* 40, 637–643 (2001)
11. Ghiglia, D.C., Mastin, G.A., Romero, L.A.: Cellular automata method for phase unwrapping. *J. Opt. Soc. Am.* 4, 267–280 (1987)
12. Servin, M., Cuevas, F.J., Malacara, D., Marroquin, J.L., Rodriguez-Vera, R.: Phase unwrapping through demodulation using the RPT technique. *Applied Optics* 38, 1934–1940 (1999)
13. Cuevas, F.J., Servin, M., Stavroudis, O.N., Rodriguez-Vera, R.: Multi layer neural network applied to phase and depth recovery from fringe patterns. *Optics Communications* 181, 239–259 (2000)
14. Cuevas, F.J., Servin, M., Rodriguez-Vera, R.: Depth recovery using radial basis functions. *Opt. Commun.* 163, 270–277 (1999)
15. Cuevas, F.J., Sossa-Azuela, J.H., Servin, M.: A parametric method applied to phase recovery from a fringe pattern based on a genetic algorithm. *Opt. Commun.* 203(3-6), 231–239 (2002)
16. Mancilla, L.E., Carpio, J.M., Cuevas, F.J.: Demodulation of Interferograms of Closed Fringes by Zernike Polynomials using a technique of Soft Computing. *Engineering Letters* 15(1), 99–104 (2007)
17. Toledo, L.E., Cuevas, F.J.: Optical Metrology by Fringe Processing on Independent Windows Using a Genetic Algorithm. *Experimental Mechanics* 48, 559–569 (2008)

18. Cuevas, F.J., Mendoza, F., Servin, M., Sossa-Azuela, J.H.: Window fringe pattern demodulation by multi-functional fitting using a genetic algorithm. *Opt. Commun.* 261, 231–239 (2006)
19. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proc. IEEE Intl. Conf. on Neural Networks*, vol. 4, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
20. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory controllers. In: *Proc. Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43. IEEE Service Center, Nagoya (1995)
21. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on System, Man, and Cybernetics SMC* 9(1) (1979)
22. Haralick, R.M., Shapiro, L.G.: *Computer and Robot Vision*, vol. 1. Addison-Wesley (1992)
23. Rosenfeld, A., Pfalz, J.L.: Sequential Operations in Digital Picture Processing. *Journal of the ACM* 13(4), 471–494 (1966)

# Evolving Conspicuous Point Detectors for Camera Trajectory Estimation

Daniel Hernández, Gustavo Olague\*, Eddie Clemente, and León Dozal

**Abstract.** The interaction between a visual system with its environment is studied in terms of a purposive vision system with the aim of establishing a link between perception and action. A system that performs visuomotor tasks requires a selective perception process in order to execute specific motion actions. This combination is understood as a visual behavior. This paper presents a solution to the process of synthesizing visual behaviors through genetic programming, resulting in specialized visual routines that are used to estimate the trajectory of a camera within a vision based simultaneous localization and map building system. Thus, the experiments were carried out with a real-working system consisting of a robotic manipulator in a hand-eye configuration. The main idea is to evolve a conspicuous point detector based on the concept of an artificial dorsal stream. The results on this paper show that it is in fact possible to find key points in an image through a visual attention process in combination with an evolutionary algorithm to design specialized visual behaviors.

## 1 Introduction

The relationship between perception and action is the focus of the field of Active Vision. It studies how vision based systems manipulate their visual information to

---

Daniel Hernández · León Dozal

CICESE, EvoVision Project, Computer Science Department

e-mail: [dahernan@cicese.edu.mx](mailto:dahernan@cicese.edu.mx), [leon.dozal@gmail.com](mailto:leon.dozal@gmail.com)

Gustavo Olague

CICESE, Carretera Ensenada-Tijuana No.3918, Zona Playitas, Ensenada, B.C., México

e-mail: [olague@cicese.mx](mailto:olague@cicese.mx)

Eddie Clemente

Tecnológico de Estudios Superiores de Ecatepec, Avenida Tecnológico S/N,

Esq. Av. Carlos Hank González, Valle de Anáhuac, Ecatepec de Morelos

e-mail: [eddie.clemente@gmail.com](mailto:eddie.clemente@gmail.com)

\* Corresponding author.

solve a given task; as a result, from the interaction of the system with its environment. An important characteristic of an active vision system is related to its capability of changing the parameters of the camera in a purposive manner depending on the conditions of the environment [9]. Hence, a purposive vision system should be considered as part of a bigger system that interacts with its environment in a very specific manner, rather than seeing it as an isolated process ([1, 2]). The idea studied in this paper is that a purposive visual system could be integrated in a more complex system; whose interaction with the environment is performed in a specific way opposed to a general manner [10]. Thus, the objective of this paper is to synthesize a visual behavior through genetic programming with the aim of adapting a visual routine to a specific purposeful task. This research is part of a bigger area known as sensor planning where genetic algorithms have been applied successfully, see [15, 14].

The idea for this work is that a visual behavior uses specific information from the environment in order to solve a given task. The analogy is that a visual behavior performed by a person that executes a set of actions in relation to the information extraction process that is required to located itself within a map. This extraction operation is closely related to the focus of attention process, in the way that both processes seek to identify prominent landmarks in the environment that can be used as position references. Moreover, for those processes is necessary to make a selection of required features that will be useful to identify an object within the scene. Thus, a visual behavior is represented by the set of actions performed by a machine in order to achieve goals like, feature extraction and object recognition, which are useful for solving the self-localization task within a specific environment. This analogy inspired the proposed system, whose process is normally modeled as an estimation task. In other words, the system performs an action based on the information captured by the perception mechanism with the purpose of solving the self-localization task. In this system, the visual routine is accomplished with a conspicuous point detector and the action is executed by the SLAM method. In this way, the objective is to synthesize a specialized detector to estimate the trajectory of the camera for a particular path evaluated through an SLAM system in a purposeful manner. Therefore, the aim of this paper is to experimentally show that it is possible to evolve a specialized visual routine, based on a visual attention model for the trajectory estimation of a camera mounted on a robot.

## 1.1 *Visual Behavior*

In the field of behavioral robots, a behavior is defined as the set of interactions between a robot and its environment. These interactions consist of a perception-action cycle where every action depends on the state of the system and its perception of the environment. These actions change the system's future perceptions and hence change its future actions [9]. In this work, we propose to define a visual behavior as a perception-action cycle whose actions represent an information selection

process rather than reactive motion-action commands. Therefore, a visual behavior corresponds to the manner in how the system selects, from the captured visual information, the necessary information to solve the task at hand.

This paper is organized as follows. First, the concept of visual behavior is reviewed by describing three parts: 1) the conspicuous point detection based on the artificial dorsal stream model; 2) the simultaneous localization and map building system; and 3) specialization of the visual behavior through genetic programming. Secondly, the multiobjective evolutionary algorithm is described. Finally, results of a real working system are presented followed by a conclusion.

## 2 Evolutionary Visual Behavior

The main objective in this work is the synthesis of visual routines through an evolutionary algorithm. The core functionality of the visual routine is a conspicuous point detector based on a visual attention model. This routine should be adapted to work within a SLAM system in order to solve the camera trajectory estimation problem. Thus, in this section the three main aspects of our work are described: the definition of a conspicuous point detector; followed by the description of the SLAM system that is used as a testbed for the visual routines. Finally, the optimization process that creates the visual routines through an evolutionary process is explained.

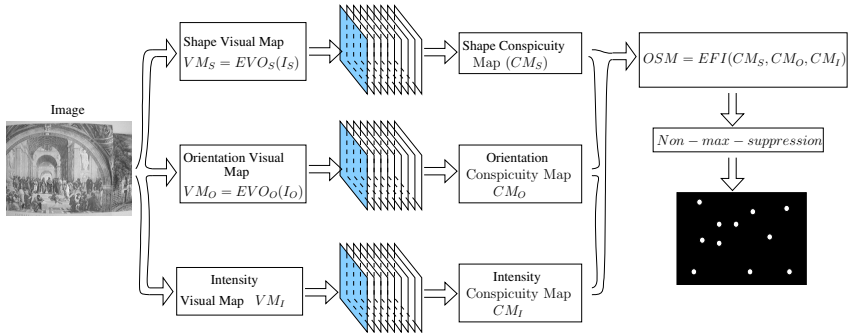
### 2.1 Conspicuous Point Detection

The concept of conspicuous point detection used in this work was developed to reformulate the theory behind the process of interest point detection. The work of Olague and Trujillo [16, 19] showed that it is in fact possible to evolve general purpose interest point detectors through genetic programming. The goal in this work is to evolve specialized visual behaviors, while at the same time maintaining general purpose properties like, *repeatability* and *dispersion*, see Figure 1.

The concept of an interest point is defined as a small region on an image which has visually significant information. In this way, the importance of a pixel is the result of applying the transformation known as *operator* and defined by  $K : R^+ \rightarrow R$ . The present paper proposes to change the computational process that estimate the importance of a pixel, and is evaluated through the concept of visual attention. Hence, the proposed conspicuous point detector work as follows.

- (i) The first step is to apply an evolved artificial dorsal stream algorithm to obtain a saliency map that is used as an interest image  $I^*$
- (ii) The second idea is to continue with the normal process of interest point detection by performing a non-maximum suppression operation.
- (iii) Finally, the third step is to establish a threshold to determine if a given maximum should be considered as a conspicuous point.





**Fig. 1** The artificial dorsal stream was used as a model to program the stages of the conspicuous point detection algorithm that is applied to obtain the optimized saliency map

### 2.1.1 Artificial Dorsal Stream

The process of computing the interest image for the conspicuous point detection is biologically inspired by the model of visual attention presented by Treisman and Gelade [26], known as an *Artificial Dorsal Stream* (ADS). The input for an ADS is an image and its output is an *Optimized Saliency Map* (OSM). An OSM is an image whose pixel values represent the saliency of a point along the considered dimensions. In this way, the ADS is divided in two main stages, feature acquisition and feature integration, see Figure 1. In the first stage, the input image is processed at three different and independent dimensions: intensity, orientation and shape. A visual operator is applied to the image creating one *Visual Map* (VM) per dimension. These maps represent the prominence of each pixel according to the corresponding visual feature. During the second stage it is necessary to combine the VMs into a single map. In this sense, the output of the ADS is an optimized saliency map (OSM) that expresses the prominence of each pixel in the input image. The following subsections describe in more detail the two stages together with the functions inside them.

### 2.1.2 Feature Acquisition Stage

As mentioned earlier, this stage consists of three visual operators whose aim is to point out the image prominences in three independent dimensions: intensity, orientation and shape. In previous work, see [12, 11], the visual attention process was defined with operators for these dimensions following a data driven approach. In this work, we follow the function driven approach of Dozal *et al.* [6] and Clemente *et al.* [4] in order to enhance the interest point detection to a paradigm for visual attention. In this way, the proposed system consists of a set of functions that are optimized through an evolutionary process to match the functionality of the dorsal stream. This leads to a set of *Evolutionary Visual Operators* that are described next.

### 2.1.3 Orientation

The detection of features along the orientation dimension are represented through the function  $EVO_O : I \rightarrow VM_O$ ; that seeks to highlights edges that seem to be interesting. The value of the  $VM_O$  correspond to the prominence of a certain pixel computed with the operator  $EVO_O$ . Table 1 shows the sets used by the evolutionary algorithm to build the operators for the orientation dimension. In this case,  $I$  is the input image,  $I_{T_O}$  are the elements in the terminal set  $T_O$ ; or the output the elements of the function set  $F_O$ ;  $D_u$  represents the image derivatives in the direction  $u \in \{x, y, xy, xx, yy\}$ , and  $G_\sigma$  corresponds to a Gaussian filtering with  $\sigma$  as the standard deviation.

**Table 1** Functions and terminals used to build the population for the orientation feature

$F_O = \{+,   +  , -,   -  ,  I_{T_O} , \times, \div, I_{T_O}^2, \sqrt{I_{T_O}}, \log_2(I_{T_O}), \frac{I_{T_O}}{\sigma}, D_x, D_y, G_{\sigma=1}, G_{\sigma=2}\}$
$T_O = \{I, D_x(I), D_{xx}(I), D_{xy}(I), D_{yy}(I), D_y(I), G_{\sigma=1}(I), G_{\sigma=2}(I)\}$

### 2.1.4 Shape

The system uses the function  $EVO_S : I \rightarrow VM_S$  to accomplish the detection of features for the shape dimension. The aim of this operator is to accentuate interesting points based on the appearance and structure of the objects in the image using a set of operators of mathematical morphology. The elements that were chosen to build the operator for this dimension are shown on Table 2.

**Table 2** Functions and terminals applied to build the population for the shape feature

$F_S = \{+, -, \times, \div, round(I_{T_S}), floor(I_{T_S}), ceil(I_{T_S}), dilation_{diamond}(I_{T_S}), dilation_{square}(I_{T_S}), dilation_{disk}(I_{T_S}), erosion_{diamond}(I_{T_S}), erosion_{square}(I_{T_S}), erosion_{disk}(I_{T_S}), skeleton(I_{T_S}), boundary(I_{T_S}), hit - miss_{diamond}(I_{T_S}), hit - miss_{square}(I_{T_S}), hit - miss_{disk}(I_{T_S}), top - hat(I_{T_S}), bottom - hat(I_{T_S}), open(I_{T_S}), close(I_{T_S})\}$
$T_S = \{I\}$

### 2.1.5 Conspicuity Maps

Now that the system created three VMs using the input image, the described operators, and the intensity dimension; the conspicuous maps (CMs) are obtained via a center-surround function  $CS : VM \rightarrow CM$ , whose goal is to emulate the center-surround receptive fields found in the natural dorsal stream. Therefore, to achieve this functionality, the system creates a prism  $VM_I(\alpha)$  with nine levels, each at a

different spatial scale  $\alpha = \{1, 2, \dots, 9\}$ . In this way, an across-scale subtraction  $\ominus$  is performed, leading to a set of center-surround maps  $VM_l(\omega)$  in such a way that the value of a pixel increases according to the contrast along its neighbors at different scales  $\omega = \{1, 2, 3, 4, 5, 6\}$ . Finally, the  $VM_l(\omega)$  maps are merged using an across-scale addition  $\oplus$  in order to obtain a conspicuous map  $CM_l$  per feature dimension. Thereafter, the system combines the  $CMs$  using the feature integration operator in order to obtain a single saliency map. This procedure is explained in the following section.

### 2.1.6 Feature Integration Stage

In this stage, the system must combine the resulting  $CMs$  in order to create a single representation. This operation is known as *Feature Integration*, that is defined as  $RFI : CM_l \rightarrow OSM$ , where  $CM_l$  are the conspicuous maps along the three dimensions  $l = \{O, S, I\}$ . In the natural system, a explicit description of this process is yet to be found, and it is uncertain how the brain makes the  $CMs$  integration, and in which region the saliency map is conceived. Nevertheless, the idea in this work is that the visual routine could be created in a purposive manner following an active vision scenario. Therefore, a saliency map is built using an *Evolved Feature Integration* (EFI) function, allowing the artificial evolution to determine a good way to combine the maps. The set of functions  $F_{fi}$  and terminals  $T_{fi}$  that are used by the evolutionary algorithm to obtain these operators can be found in Table 3.

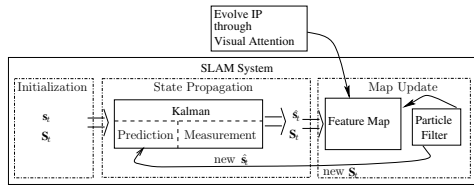
**Table 3** Functions and terminals used to build the population for the feature integration stage

$F_{fi} = \{+,  +,  -,  -,  I_{T_{fi}} , \times, \div, I_{T_{fi}}^2, \sqrt{I_{T_{fi}}}, \log_2(I_{T_{fi}}), D_x, D_y, G_{\sigma=1}, G_{\sigma=2}\}$
$T_{fi} = \{CM_O, CM_I, CM_S, D_x(CM_O), D_{xx}(CM_O), D_{xy}(CM_O), D_{yy}(CM_O), D_y(CM_O), \dots\}$

The resulting OSM is a combination of the  $CMs$ , whose pixel values represent the prominence of each pixel. Therefore, we propose to use the OSM as an interest image for the feature detection algorithm. Since, the operation is quite different from the typical operator used for interest point detection, we will refer to the points discovered by this methodology as Conspicuous Points (CPs).

## 3 Trajectory Estimation with a SLAM System

In this work, the objective of the visual behaviors is to estimate the trajectory described by the camera mounted on a robotic manipulator. In order to accomplish the task, the conspicuous point detector (CPD) is integrated within a simultaneous localization and map building (SLAM) system. This section describes how the system works and the relationship with the CPD. The objective for the behavior is



**Fig. 2** This figure depicts the stages of the SLAM system along the evolutionary method integrated within the map update stage

to highlight useful information perceived by the images in such a way of finding the position of the camera over time. The SLAM system is used to evaluate the efficiency of the visual behaviors in terms of a trajectory estimation problem and is based on the MonoSLAM system presented by Davison *et al.* [5]. The system uses a single camera and computes a feature based map of the environment in order to create a representation. The localization process is performed as a state estimation task through the Kalman filtering method. The map construction is obtained using the concept of conspicuous points to find prominent visual landmarks in the environment [13]. The SLAM process is divided into three stages: initialization, state propagation, and map update.

**Initialization Stage.** To estimate the trajectory of the camera, the system must calculate its position while it moves; therefore, the state of the SLAM system is defined by the camera’s pose and speed. In this way, the Kalman filtering process estimates the attitude of the camera  $\hat{x}_v$  given by its position  $r$  and orientation  $q$ , and linear and angular speeds,  $v$  and  $\omega$ . Along this, the system needs to maintain a map of the environment; this is accomplished through a sparse set of visual landmarks. For this reason, the system should track the position of the camera and the spacial position of each feature  $y_i$ . Hence, the system estimates the state  $s_t$ , composed of the information about the camera and the features. Then, the state estimation  $\hat{s}_t$  is coupled with the estimation uncertainty  $S_t$ . Thus, the system starts, with a known initial state; in other words, the camera position is known using a starting map made of four landmarks. The state  $s_t$  and its covariance  $S_t$  are defined as follows.

$$s_t = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{pmatrix} \quad S_t = \begin{bmatrix} S_{xx} & S_{xy_1} & S_{xy_2} & \dots \\ S_{y_1x} & S_{y_1y_1} & S_{y_1y_2} & \dots \\ S_{y_2x} & S_{y_2y_1} & S_{y_2y_2} & \dots \\ \vdots & \vdots & & \end{bmatrix}$$

**State Propagation.** The state estimation is computed with a Kalman filtering method, which is divided in two stages, prediction and measurement. The future states of the camera are assumed to follow the dynamic model through the following relationship.

$$s_t = \mathbf{A}s_{t-1} + \mathbf{w}_t,$$

where  $\mathbf{A}$  is the state transition matrix, and  $\mathbf{w}_t$  is the process noise. At each time step, the filter makes a *prediction* of the current state based on the previous state and the dynamic model. The prediction  $s_t^-$  is also known as the *a priori* state; and together with its error matrix  $\mathbf{S}_t^-$  this first step is called the *prediction stage*. Thus, the dynamic model is defined by

$$\begin{aligned} s_t^- &= \mathbf{A}s_{t-1} \\ \mathbf{S}_t^- &= \mathbf{A}\mathbf{S}_{t-1}\mathbf{A}^T + \lambda_w, \end{aligned}$$

where  $\lambda_w$  is the process covariance noise assumed to be defined as a white Gaussian noise.

*Measurement stage.* This stage consists in using the images captured by the camera to improve the state estimation. Thus, the measurements  $\mathbf{z}_t$ , about the image location at time  $t$  of some visual landmarks, are assumed to be related to the current state of the camera by

$$\mathbf{z}_t = \mathbf{C}\mathbf{s}_t + \mathbf{v}_t$$

where  $\mathbf{v}_t$  is the measurement noise, and  $\mathbf{C}$  describes the camera pose through the position of the visual landmarks within the images. The measurement model is then used to establish an *a posteriori* state estimate  $\hat{\mathbf{s}}_t$  along its error matrix  $\mathbf{S}_t$  by incorporating the measurements  $\mathbf{z}_t$  with the following equations

$$\begin{aligned} \hat{\mathbf{s}}_t &= s_t^- + \mathbf{G}_t(\mathbf{z}_t - \mathbf{C}s_t^-) \\ \mathbf{S}_t &= \mathbf{S}_t^- - \mathbf{G}_t\mathbf{C}\mathbf{S}_t^- \end{aligned}$$

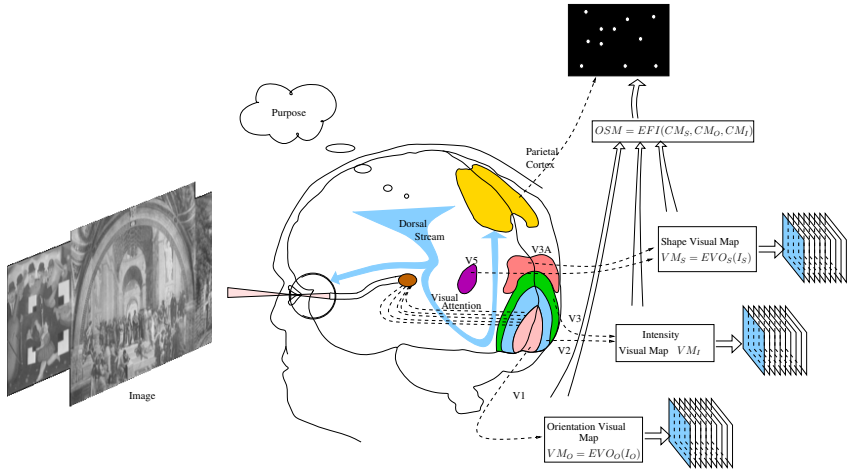
where  $\mathbf{G}_t$  is the variable known as Kalman gain, defined by

$$\mathbf{G}_t = \mathbf{S}_t^- \mathbf{C}^T (\mathbf{C}\mathbf{S}_t^- \mathbf{C}^T + \lambda_v)^{-1}$$

with  $\lambda_v$  being the measurement covariance. In this manner, the *a posteriori* state estimation at time  $t$  is complete; therefore, the state  $\hat{\mathbf{s}}_t$  together with its uncertainty matrix  $\mathbf{S}_t$  is ready for the next estimation cycle.

**Map Update.** Now that the system has a good estimation of the camera position, it necessary to capture more information of the environment. This is performed by extending the feature map. In this way, the system employs the conspicuous point detection algorithm to find landmarks in the environment that may improve the state estimations in future frames. In this way, the designed detector must bring implicitly the required characteristics through the conspicuous points in order to help the system to find a solution to solve the position estimation task. The final step devoted to the map update calculates the depth of the visual landmarks. Since, the system uses monocular vision, it applies a particle filter to find the depth of the visual landmarks. Once, the depth is know for a given point; thus, the information is added to the map for further estimations.

After these three steps, the estimation cycle of the Kalman filter is complete, and the position estimation is ready for the next time step. The cycle is executed for each image captured by the camera.



**Fig. 3** The stages of the SLAM system along the evolutionary method integrated within the map update stage

### 4 Synthesis of Visual Behaviors

This section is dedicated to describe the evolutionary process for the ADS using the organic genetic programming (OGP) [6, 4]. In the proposed model the genotype is composed of three operators, which can be seen as the genes of a complex chromosome. Figure 1 depicts the corresponding phenotype. In this way, the ADS helps in the process of extraction of the conspicuous points. For this reason, all operators inside the detection algorithm should be seen as a single entity and not as independent operations. The proposed representation has the objective of developing the design of complex functions capable of solving the task at hand. In this case, the construction of a conspicuous point detector to be used inside the SLAM system. Thus, it should be noted that each individual of the population represents a complete ADS with the purpose of extracting conspicuous points; hence, it is not just a list of tree-based operations, but rather a complex information processing system as shown in Figure 1. The evolved detector is used inside the previously described localization process shown in 3. Due to the complexity of the problem, the evolution was carried out in a simple but still representative motion path. In this case, the camera is moving on a straight line, parallel to a wall rich in visual information. The SLAM system determines the camera position using the information provided by the evolved detector and the map building stage. Thus, the resulting position estimations evaluate the performance of the detector according to the trajectory estimation task. The evolved detectors should have

the properties mentioned in section 2.1. Due to the computational effort required by the evolutionary algorithm, an image sequence was captured during the camera motion in order to evolve the detectors with an off-line evaluation process. This step could be easily replaced in our proposed system. Also, one of the hypothesis in this work is that a good detector evolved for a tracking task, is that that achieves high repeatability and high point dispersion, that leads to finding stable environment landmarks and will help to create a disperse feature map. Since, we have considered several objectives for the evolutionary process; the evolved detectors are evaluated through the following fitness functions with the aim of maximizing the repeatability and point dispersion, while minimizing the trajectory estimation error. Thus, for simplicity we normalize the functions before applying the minimization process.

**Repeatability.** The average repeatability  $r_K(\varepsilon)$  is calculated for the operator  $K$  between two consecutive images using a neighborhood of size  $\varepsilon$ . It is important to note that the repeatability is calculated through the position of the camera using a highly-accurate robot movement. This test replace the application of the homographies between images.

$$rI_i(\varepsilon) = \frac{|R_i(\varepsilon)|}{\min(\gamma_{i-1}, \gamma_i)}$$

where  $\gamma_{i-1} = |\{x_{i-1}^c\}|$  and  $\gamma_i = |\{x_i^c\}|$  are the number of points detected in images  $I_{i-1}$  and  $I_i$ .  $R_i(\varepsilon)$  is a set of pairs of points  $(x_{i-1}^c, x_i^c)$  that were found in two consecutive images within a region of radius  $\varepsilon$ :

$$f_1 = \frac{1}{r_K(\varepsilon) + c_1}$$

where  $c_1$  is a constant to avoid an invalid division.

**Dispersion.**  $\mathcal{D}_p(K)$  is the average dispersion of the located points within the image sequence using operator  $K$ ; where  $c_2 = 10$  is a normalization constant.

$$f_2 = \frac{1}{e^{\mathcal{D}_p(K) - c_2}}$$

The point dispersion in image  $I_i$  is calculated using the points's entropy  $D(I, X) = -\sum P_j \cdot \log_2(P_j)$  where  $X$  is the set of detected points and  $P_j$  is approximated using a histogram.

**Trajectory Error.** This fitness is measured using the mean squared error of the estimated trajectory using a real straight-line trajectory.

$$f_3 = \sum_{i=1}^M \frac{[x_{v_i} - \hat{x}_{v_i}]^2}{M}$$

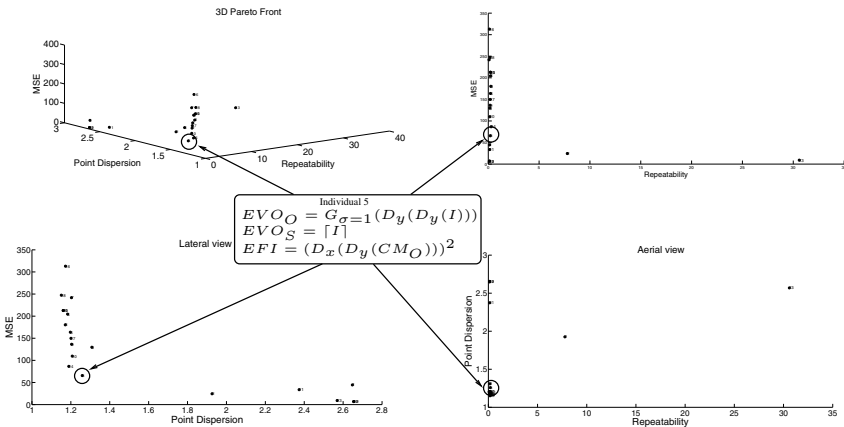
The parameters used for the evolutionary process can be seen in Table 4, since this is described in terms of a multiobjective process the SPEA2 [20] algorithm was used to find the optimal Pareto front.

**Table 4** Parameters of the Multi-objective GP for the synthesis of conspicuous point detectors

Parameters	Description
Population	20 individuals
Generations	20 iterations
Initial population	Ramped Half-and-Half
Genetic operations probabilities	Crossover $p_c = 0.85$ Mutation $p_\mu = 0.15$
Max three depth	6 levels
File size (SPEA2)	8
Parent selection	8

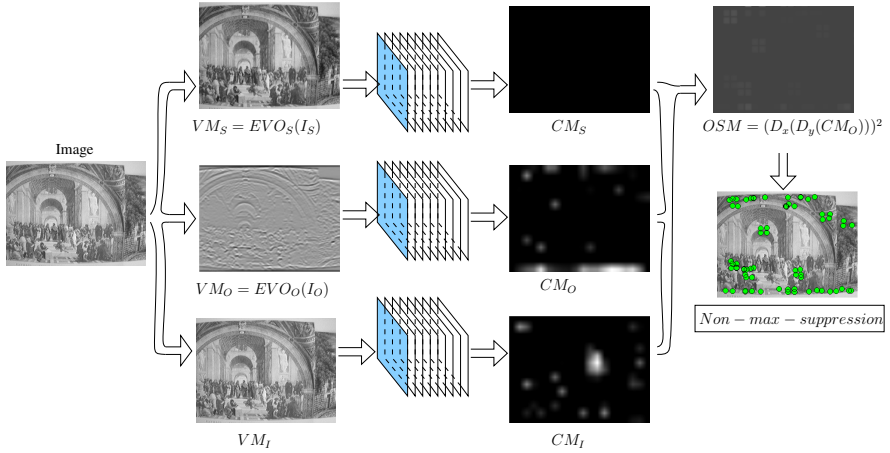
### 5 Experimental Results

This section describes the experiments in order to outline the implemented system that synthesizes an artificial dorsal stream using a conspicuous point detection algorithm. The resulting population of Pareto fronts can be seen in Figure 4, which shows the distribution of the best individuals along the fitness space. The Table 5 lists the non-dominated individuals forming the Pareto front that were produced with the system just described. The resulting operators are called conspicuous point detectors for SLAM (CPSLAM). The individuals were tested inside a monocular vision SLAM system. All experiments were performed in a Dell Precision T7500 workstation, Intel Xeon 8 Core, NVIDIA Quadro FX 3800 running OpenSUSE 11.1 linux operating system. Figure 6 shows the execution of the system using the individual CPSLAM5 that achieves the better results. The sequence on the left correspond to the captured images where the ellipses on the images represent the computed visual landmarks. The right side of the figure depicts the error on the estimated trajectory for the XY and XZ planes.



**Fig. 4** Resulting Pareto front for the evolutionary process



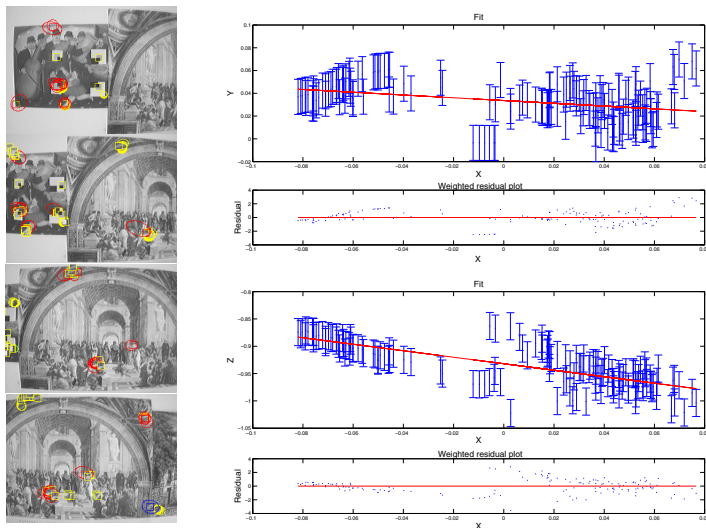


**Fig. 5** This figure shows the *CPSLAM5* individual that was obtained from the Pareto and that achieves optimal conspicuous points

**Table 5** Individuals discovered with the evolutionary process that create the Pareto front of Figure 4

Name	Operator	Fitness
CPSLAM1	$EVO_O = \text{round}(D_x(D_y(I))) + 0.09$ $EVO_S = \text{hit} - \text{miss}_{\text{diamond}}(I)$ $EFI = \frac{CM_I}{D_x(CM_S)}$	$f_1 = 0.124$ $f_2 = 2.647$ $f_3 = 44.51$
CPSLAM2	$EVO_O = \text{thresh}(\log_2(\lfloor \text{infimum}(D_x(D_x(I)), I) \rfloor))$ $EVO_S = \text{dilate}_{\text{square}}(\frac{I}{0.97})$ $EFI = \lfloor \frac{(CM_I - D_x(D_x(CM_I)))}{0.63} \rfloor$	$f_1 = 7.774$ $f_2 = 1.926$ $f_3 = 24.459$
CPSLAM5	$EVO_O = G_{\sigma=1}(D_x(D_y(I)))$ $EVO_S = \lceil I \rceil$ $EFI = (D_x(D_y(CM_O)))^2$	$f_1 = 0.195$ $f_2 = 1.258$ $f_3 = 65.67$
CPSLAM11	$EVO_O = \text{infimum}(I^2, \text{infimum}(D_x(\text{supremum}(D_y(I), D_x(I)/2)))^2,  D_x(D_x(I)) - (G_{\sigma=1} * I) )$ $EVO_S = k\text{Sust}(\lfloor \text{Erode}_{\text{diamond}}(\text{Dilate}_{\text{square}}(I)) \rfloor, 0.05)$ $EFI = \text{round}((D_y(D_y(CM_I)))/0.87)$	$f_1 = 0.134$ $f_2 = 2.373$ $f_3 = 33.67$
CPSLAM13	$EVO_O = \sqrt{\text{thresh}(\text{supremum}((D_x(D_x(I)))^{\frac{1}{0.85}}, \text{round}(I)))}$ $EVO_S = \text{hit} - \text{miss}_{\text{diamond}}(\text{perimeter}((I/0.97) + (I + 0.55))^{\frac{1}{0.85}})$ $EFI = (\text{round}(D_y(D_y(CM_I)))/0.87)^2$	$f_1 = 30.592$ $f_2 = 2.569$ $f_3 = 9.164$
CPSLAM14	$EVO_O = \sqrt{\lfloor  D_x(I) - \lceil D_y(D_y(I)) \rceil \rfloor}$ $EVO_S = \text{hit} - \text{miss}_{\text{diamond}}(\text{perimeter}(I/0.97) + (I + 0.55))^{\frac{1}{0.85}})$ $EFI = \lfloor  e^{D_x(D_x(CM_S)) - CM_S} - (D_y(D_y(CM_I)))/0.87  \times G_{\sigma=2} * (D_x(D_x(CM_S)) - D_x(D_x(CM_O))) + ((CM_O + D_x(CM_I)) - G_{\sigma=1} * D_x(D_x(CM_S))) + 0.18 \rfloor$	$f_1 = 0.279$ $f_2 = 1.189$ $f_3 = 86.504$
CPSLAM18	$EVO_O = \sqrt{\text{thresh}(\text{supremum}((G_{\sigma=1} * D_y(D_y(I)))^{\frac{1}{0.85}}, \text{round}(G_{\sigma=1} * I)))}$ $EVO_S = \text{hit} - \text{miss}_{\text{diamond}}(\lceil I \rceil + (I + 0.55))^{\frac{1}{0.85}}$ $EFI = (CM_O + D_x(D_x(CM_O)) - G_{\sigma=1} * D_x(D_x(CM_S)) + 0.18)$	$f_1 = 0.171$ $f_2 = 1.151$ $f_3 = 200.47$
CPSLAM19	$EVO_O = \text{infimum}(I^2, D_x(I))$ $EVO_S = \text{hit} - \text{miss}_{\text{disk}}(I)$ $EFI = \text{round}(D_y(D_y(CM_S)))^2$	$f_1 = 0.124$ $f_2 = 2.654$ $f_3 = 6.851$

It results interesting to note that most of the individuals do not use all the dimensions for the construction of the saliency map. This can be related to the fact that the trajectory that was used for evaluation was very simple. The dimension that was commonly applied was the orientation, which can be interpreted as the feature that is more useful for solving this particular trajectory problem due to the edge detection capabilities of the proposed operations; i.e., the case of a straight-line displacement, see Figure 5.



**Fig. 6** System execution using the *CPSLAM5* individual. Note that this individual is basically performing an interest point detection process.

## 6 Conclusions

The objective of this paper was to synthesize, through a multiobjective evolutionary approach a set of specialized visual behaviors in the form of conspicuous point detector based on an artificial dorsal stream model. The resulting behaviors are based on solving the trajectory estimation of a camera, when being used inside a real-world SLAM system. The results of this work show that it is coherent to see an artificial evolutionary method as a purposive process. Moreover, we showed that it is possible to find conspicuous points that replace interest point detection in an image through the concept of visual attention. Finally, we have shown also that it is possible to design such detectors to solve a given task.

## 7 Future work

The proposed perspective of behavioral opens an interesting and new research avenue and we give next some points that we would like to explore in the future.

- (i) As was mentioned earlier, the evolved detectors were evaluated with a simple straight-line motion path. It would be interesting to test the resulting detectors in more complex trajectories and also to evolve new detectors for other trajectories.
- (ii) The application of an evolved conspicuous point detector for other visual tasks. In particular, we would like to compare the efficiency against traditional interest point detectors.
- (iii) Also, we would like to introduce motion related visual operations to see if they improve the functionality of the detectors.
- (iv) Finally, we would like to develop a set of tests which insights about the feature dimensions that are better according to the group of invariance.

**Acknowledgements.** This research was funded by CONACyT through the Project 155045 - “Evolución de Cerebros Artificiales en Visión por Computadora”. First author supported by scholarship 267339/220773 from CONACyT. This research was also supported by TESE through the project DIMI-MCIM-004/08.

## References

1. Aloimonos, J., Weiss, I., Bandyopadhyay, A.: Active vision. In: Proceedings of the First International Conference on Computer Vision, pp. 35–54 (1987)
2. Aloimonos, Y.: Active Perception, 292 pages. Lawrence Erlbaum Associates, Publishers (1993)
3. Ballard, D.: Animate Vision. *Artificial Intelligence Journal* 48, 57–86 (1991)
4. Clemente, E., Olague, G., Dozal, L., Mancilla, M.: Object Recognition with an Optimized Ventral Stream Model Using Genetic Programming. In: Di Chio, C., Agapitos, A., Cagnoni, S., Cotta, C., de Vega, F.F., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Langdon, W.B., Merelo-Guervós, J.J., Preuss, M., Richter, H., Silva, S., Simões, A., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Togelius, J., Urquhart, N., Uyar, A.Ş., Yannakakis, G.N. (eds.) *EvoApplications 2012*. LNCS, vol. 7248, pp. 315–325. Springer, Heidelberg (2012)
5. Davison, A.J.: Real-Time Simultaneous Localisation and Mapping with a Single Camera. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 1403–1410. IEEE Computer Society, Washington, DC (2003)
6. Dozal, L., Olague, G., Clemente, E., Sánchez, M.: Evolving Visual Attention Programs through EVO Features. In: Di Chio, C., Agapitos, A., Cagnoni, S., Cotta, C., de Vega, F.F., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Langdon, W.B., Merelo-Guervós, J.J., Preuss, M., Richter, H., Silva, S., Simões, A., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Togelius, J., Urquhart, N., Uyar, A.Ş., Yannakakis, G.N. (eds.) *EvoApplications 2012*. LNCS, vol. 7248, pp. 326–335. Springer, Heidelberg (2012)

7. Dunn, E., Olague, G.: Multi-objective Sensor Planning for Efficient and Accurate Object Reconstruction. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) *EvoWorkshops 2004*. LNCS, vol. 3005, pp. 312–321. Springer, Heidelberg (2004)
8. Dunn, E., Olague, G.: Pareto Optimal Camera Placement for Automated Visual Inspection. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3821–3826 (2005)
9. Fermüller, C., Aloimonos, Y.: The Synthesis of Vision and Action. In: Landy, et al. (eds.) *Exploratory Vision: The Active Eye*, ch. 9, pp. 205–240. Springer (1995)
10. Hernández, D., Olague, G., Clemente, E., Dozal, L.: Evolutionary Purposive or Behavioral Vision for Camera Trajectory Estimation. In: Di Chio, C., Agapitos, A., Cagnoni, S., Cotta, C., de Vega, F.F., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Langdon, W.B., Merelo-Guervós, J.J., Preuss, M., Richter, H., Silva, S., Simões, A., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Togelius, J., Urquhart, N., Uyar, A.Ş., Yannakakis, G.N. (eds.) *EvoApplications 2012*. LNCS, vol. 7248, pp. 336–345. Springer, Heidelberg (2012)
11. Itti, L., Koch, C.: Computational modelling of visual attention. *Nature Review Neuroscience* 2(3), 194–203 (2001)
12. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol* 4(4), 219–227 (1985)
13. Lepetit, V., Fua, P.: Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. In: *Foundations and Trends in Computer Graphics and Vision*, vol. 1, pp. 1–89 (2005)
14. Olague, G.: Automated Photogrammetric Network Design using Genetic Algorithms. *Photogrammetric Engineering & Remote Sensing* 68(5), 423–431 (2002)
15. Olague, G., Mohr, R.: Optimal Camera Placement for Accurate Reconstruction. *Pattern Recognition* 27(4), 927–944 (2002)
16. Olague, G., Trujillo, L.: Interest Point Detection through Multiobjective Genetic Programming. *Applied Soft Computing* (to appear, 2012)
17. Shi, J., Tomasi, C.: Good features to track. In: *Proceedings of Computer Vision and Pattern Recognition*, pp. 593–600 (1994)
18. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. *Cognitive Psychology* 12(1), 97–136 (1980)
19. Trujillo, L., Olague, G.: Automated Design of Image Operators that Detect Interest Points. *Evolutionary Computation* 16, 483–507 (2008)
20. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm. *Technical report, Evolutionary Methods for Design* (2001)

# Purposive Evolution for Object Recognition Using an Artificial Visual Cortex

Eddie Clemente, Gustavo Olague\*, and Leon Dozal

**Abstract.** This work presents a novel approach to synthesize an artificial visual cortex based on what we call organic genetic programming. Primate brains have several distinctive features that help in the outstanding display of perception achieved by the visual system, including binocular vision, memory, learning, and recognition, to mention but a few. These features are processed by a complex arrangement of highly interconnected and numerous cortical visual areas. This paper describes a system composed of an artificial dorsal pathway, or where stream, and an artificial ventral pathway, or what stream, that are fused to create a kind of artificial visual cortex. The idea is to show that genetic programming is able to evolve a high number of heterogeneous trees thanks to the hierarchical structure of our virtual brain. Thus, the proposal uses two key ideas: 1) the recognition of objects can be achieved by a hierarchical structure using the concept of function composition, 2) the evolved functions can be related to the tissues of an artificial organ. Experimental results provide evidence that high recognition rates could be achieved for a well-known multiclass object recognition problem.

## 1 Introduction

The brain is the most sophisticated organ in the human body; its fundamental task is to control and manage the activities that perform sensorial organs. The neurologists have divided the human brain into four lobes: frontal, temporal, parietal, and

---

Eddie Clemente

Tecnológico de Estudios Superiores de Ecatepec, Avenida Tecnológico S/N,  
Esq. Av. Carlos Hank González, Valle de Anáhuac, Ecatepec de Morelos  
e-mail: [eddie.clemente@gmail.com](mailto:eddie.clemente@gmail.com)

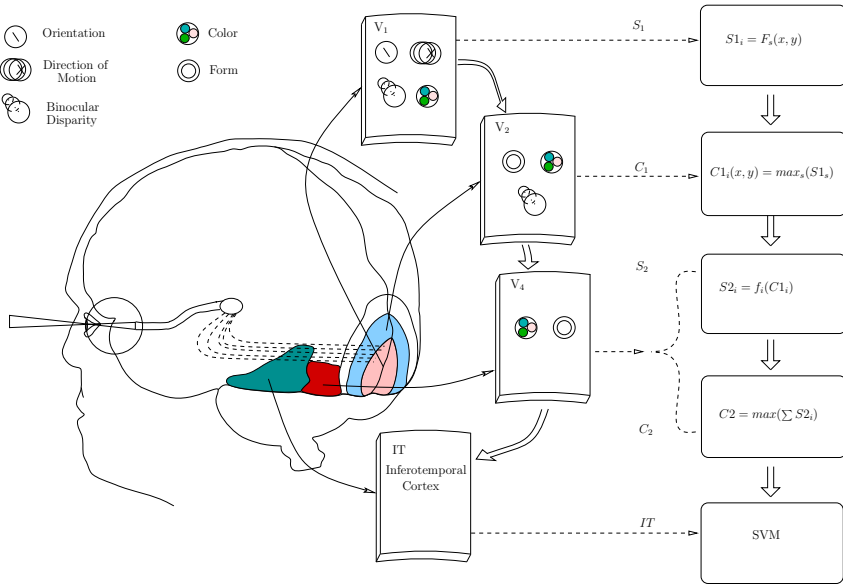
Gustavo Olague

CICESE, EvoVision Project, Applied Physics Division  
e-mail: [olague@cicese.mx](mailto:olague@cicese.mx)

Leon Dozal

CICESE, EvoVision Project, Computer Science Department  
e-mail: [leon.dozal@gmail.com](mailto:leon.dozal@gmail.com)

\* Corresponding author.



**Fig. 1** Analogy between the ventral stream and the proposed computational model

occipital. The last one has special interest to the research community interested in the sense of vision, because this is the lobe where it is located the visual cortex. In fact, the primary visual cortex and secondary visual areas are specialized for image processing, object localization, and the estimation of direction, velocity and object trajectories. Figures 1 illustrates the main ideas that we are proposing to approach the problem of object recognition. We divide the approach into two key ideas. The first one is related to the identification of salient features through the application of a set of functions that should be able to identify the salience properties that characterize a given object. In general, the works that follow a modeling of the human visual system like [8], [24], [18], [15], [11] are based on a set of image patches that are used as a dictionary of visual words. These small images represent the most common and useful characteristics presented in all images of a database integrated by a number of visual categories. The hypothesis made in our work is that such set of image patches could be substituted with a set of mathematical functions. The second idea, that we want to outline is based on the concept of an organ. In biology, an organ is described as a collection of tissues joined in a structural unit to serve a common function. In particular, we are interested in studying the brain; specially the visual cortex and how it is explained the functionality of the main tissues involved into the process of object recognition.

The goal of this work is to outline a methodology based on organic genetic programming implemented through the modeling of the hierarchical structure of the

visual cortex and the concept of function composition inspired from the idea of an organ. In this way, a functional approach is enforced in order to solve the problem of object recognition.

## 2 Visual Cortex

Visual processing is performed by the brain, and the explanation about how it works is based on the idea of two visual subsystems. Today, it is widely accepted the two-streams hypothesis as a way of describing the phenomenon of visual perception. This knowledge is based on neuropsychological, neurophysiological, and psychophysical evidence regarding the existence of two visual subsystems known as ventral and dorsal streams. Thus, the explanation is founded on the idea that both systems manage the same visual information, but the difference lies in the transformations that both streams performed to the visual data. This is clearly exposed in the change of paradigm from a what/where dichotomy into a vision-for-action/vision-for-perception duality used to explain the same dorsal/ventral anatomical distinction, see [22], [23], [29], [28], [17]. Next, we briefly describe the ventral and dorsal streams.

The ventral stream is largely associated with object recognition and shape representation, see [19]. The ventral or what pathway starts at the retina, and it receives its main input from the parvocellular layer of the lateral geniculate nucleus of the thalamus, and it projects into  $V1$ , which is part of the primary visual cortex, also called striate cortex, which is located at the back of the brain. Then, the ventral pathway continues into the visual areas  $V2$  and  $V4$ , which are part of a region known as the extrastriate visual cortex, and finally to the areas  $TEO$  and  $TE$  of the inferior temporal cortex. In computer science, the ventral stream is explained as performing a hierarchical and feedforward process that is specialized for object recognition and is biologically inspired from [9]. Most proposed models start with an image that is decomposed into a set of alternating “S” and “C” layers that are named after the discovery of Hubel and Wiesel of the simple and complex cells, see [9]. The idea was originally implemented by Fukushima in the neocognitron system, see [8]. This system was further enhanced by other authors including the convolutional networks [13], and the HMAX model [21]. In all these models the simple layers apply local filters in order to compute higher-order features, and the complex layers increase invariance by combining units of the same type, see [27].

The dorsal stream, also known as the “where” or “how” pathway, is related to the visual processing of spatial locations. Nevertheless, this part of the visual processing is still controversial; since, the dorsal stream is said to be involved in the guidance of actions, as well as, the spatial localization of objects in space. Like the ventral stream, the dorsal stream starts at the retina, and it receives its main, if not total, input from the magnocellular retinocortical layer of the lateral geniculate nucleus of the thalamus, and it projects into  $V1$ , but it also receives direct subcortical inputs from the superior colliculus and pulvinar structures. Then, the dorsal stream continues through  $V2$ ,  $V3$ , the middle temporal area  $MT$ , and the medial superior

temporal area *MST*, which are part of the extrastriate visual cortex; and finishing in the posterior parietal cortex and adjacent areas. In general, it is acknowledged that visual attention is performed by the dorsal stream, and the most widely accepted paradigm for visual attention is the feature integration theory, see [26]. However, there are other theories that attempts to explain the workings of visual attention in the dorsal stream, like [19] and [31]; or even a work that relate visual attention to both streams, see [6]. In computer science, the first computational approach for visual attention was introduced by Koch and Ullman in 1985, see [12]. Later, other researchers proposed several methodologies, which are based mostly in the feature integration theory, like [14], and [11]. In all these models the image is decomposed in several dimensions in order to obtain a set of conspicuity maps and then integrate them into a saliency map.

In this way, the visual system has been defined by two information processing streams organized in two broad structures subserving object and spatial vision. The classical dichotomy between object and space perception focuses on the importance of a single and general purpose representation. On the other hand, the “what” and “how” theory of Milner and Goodale [17] gives emphasis to the idea that the visual system is defined according to the requirements of the task that each stream subserves. The idea is to define multiple frames of reference giving special attention to the goal of the observer. In this way, the same object and spatial information is transformed by the visual system for different purposes. Thus, the ventral system represents the visual world in allocentric coordinates by promoting conscious perceptual awareness. On the other hand, the dorsal stream uses egocentric coordinates to transform the information about objects location, orientation and size, see [5].

### 3 Evolution and Teleology for Visual Processing

This section is devoted to the idea that visual processing is a product of brain evolution; and therefore it is plausible to follow an artificial evolutionary approach in the search of object recognition programs. The explanation that outlines our computational approach will be developed in two parts. The first reviews some explanations about how the brain has evolved. Next, we explain how the two stream hypothesis can be understood in teleological terms. In fact, we would like to stress that there are two main viewpoints that are used within evolutionary explanations. Today, there are two schools of knowledge, mechanistic and teleological, that attempt to provide an explanation for understanding nature. Note, that teleological explanations does not exclude mechanisms. Also, the controversy is still alive mainly because teleological explanations cannot materialise the idea of purpose and at the same time the mechanistic explanations cannot vanish the idea of purpose. Nevertheless, a purpose is not a desire; and when we refer to a purpose, we talk in terms if it is achieved or not. Hence, we claim that our personal teleological viewpoint offers the possibility



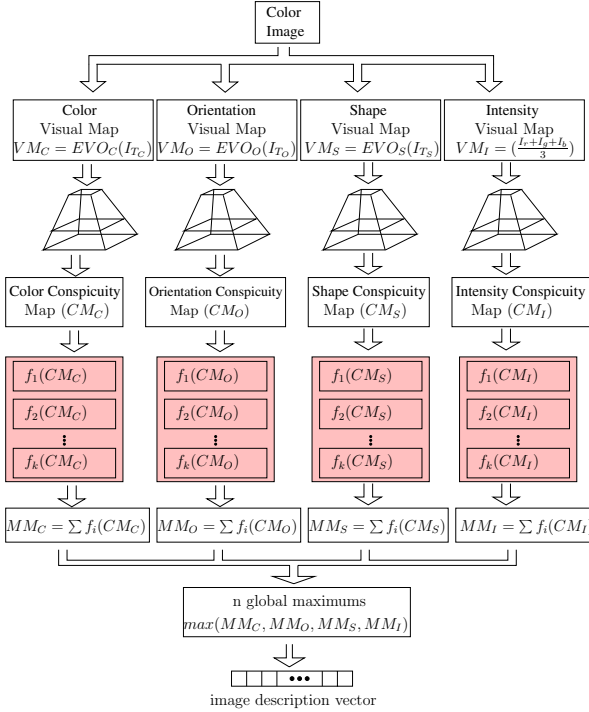
of developing a newer and rich explanation about artificial evolution, which is in accordance to many theoreticians of brain evolution that use a teleological language, see [17, 5, 2]; and philosophers of science, see [1, 14, 25].

Classical definitions of vision implicitly and explicitly assume that the purpose of the visual system is to construct some sort of internal model of the environment; in other words, a kind of visual representation of the real world, that serves as the perceptual foundation for all visually derived thought and action. The approaches to study the structure and functionality of the neocortex are based on comparative, developmental, and functional or adaptationist explanations. In particular, we hold to the tenet of many biologists that adaptation, in nature, makes the organs to suit the work they have to do; hence, developmental and functional explanations are complementary and not alternate explanations. This is clearly seen from the fact that different species from the same taxonomic group have evolved specialized visual mechanisms, which are coherent and highly correlated to particular and specific cognitive and behavioral functions, which were evolved based on the principle of natural selection. In this way, the goal in this work is to evolve a system that is based on a description of the ventral and dorsal streams and to adapt their behavior to an specific task.

#### 4 Artificial Visual Cortex (AVC)

The proposed approach is inspired and based on the idea that an organ is a collection of tissues joint in a structural unit to serve a common function. In this way, the central nervous system is understood as the organ and the study is limited to the retina, brain, in particular the visual cortex, and how it is processed the visual information. In the literature, the computational approaches inspired from the visual cortex are always centered into the dorsal or ventral streams, with the idea of solving the visual attention or object recognition tasks, respectively. Indeed, there is not yet a significant work that attempts to model the visual cortex as a whole and unique system. In general, the only works that consider the subject use a visual attention module as input to the object recognition method to create a more complex system. Instead of this simple approach and according with the reviewed literature that states that layers V1 and V2 are part of both streams, dorsal and ventral, we propose a new modeling, see Figure 2 in order to create an artificial visual cortex using the idea of function composition. This new methodology makes necessary to understand an image like the graph of a function. The function in this case is understood like the physical, geometrical, or other properties of the scene. In this way, in order to describe the idea we define an image as the graph of a function.

**Definition 1 (Image as the graph of a function)** *Let  $f$  be a function  $f : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ . The graph or image  $I$  of  $f$  is the subset of  $\mathbb{R}^3$  that consist of the points  $(x, y, f(x, y))$ , in which the ordered pair  $(x, y)$  is a point in  $U$  and  $f(x, y)$  is the value at that point. Symbolically, the image  $I = \{(x, y, f(x, y)) \in \mathbb{R}^3 | (x, y) \in U\}$ .*



**Fig. 2** Flowchart of the artificial visual cortex. Note, the similarity with the visual attention process in which the image is decomposed into several dimensions. In our approach a function driven paradigm is enforced to avoid the application of image patches.

In this way, the image is the input of a computational system that mimics the functionality of an artificial visual cortex by replicating the hierarchical structure of the natural system. Contrary to previous research devoted to object recognition, where the ventral stream is modeled through a data-driven scenario; here, the object recognition system is designed following the hierarchical structure of the dorsal and ventral streams, as well as, the idea that each layer can be modeled with a set of mathematical functions that replicate the functionality of a virtual tissue.

Genetic programming is used as the paradigm to implement the proposed approach for which a set of evolutionary visual operators (EVOs) are optimized according to the hierarchical structure being evolved in the search of an optimal object recognition program. The aim of genetic programming is to find the best set of EVOs using a number of building blocks and the whole hierarchical structure, in order to find a solution to a multi-class object recognition problem. One advantage of the functional approach compared to previous, data-driven, approaches is reflected on the lower amount of computations that brings a significant economy in the number of computer operations without sacrificing the overall quality. Next, it is described the proposed system according to the dorsal and ventral streams.

**Table 1** Set of functions and terminals used by  $EVO_C$  to create the visual map  $VM_C$

$F_C = \{+, -, \times, \div,  +,  - , \sqrt{I_{T_C}}, I_{T_C}^2, \log_2(I_{T_C}),$ $Exp(I_{T_C}, Complement(I_{T_C}))\}$
$T_C = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$

### 4.1 Artificial Dorsal Stream (ADS)

This first part of the system is based on the psychological model of Treisman and Gelade [26], which was successfully implemented in [[27], [11]]. The first step of the process is represented by the image acquired by the camera, whose natural counterpart is the retina. Here, the system considers digital color images that are composed of three images at different wavelengths of light that are red, green, and blue. Note, that it is possible to convert an image represented in RGB space into another color space. Thus, we say that a color image is the set of images named  $I_{color} = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$ . Next, four visual operators are applied separately to emphasize: intensity, color, orientation, and shape. In biological plausible models such as [11], some of these operators are established according to knowledge in neuroscience about how these features are obtained in the visual cortex of the brain, using a data-driven approach. Here, the operation of the dorsal stream is emulated by a set of functions that are evolved with genetic programming to obtain an optimal set of EVOs as depicted on Figure 2. Each EVO is represented as a specialized function that is evolved from a set of suitable characteristics that are used to create a set of visual maps. Note, that each visual map is processed by a number of functions used to create a pyramid that achieves invariance to position and scale. In fact, the result of this process is a conspicuity map for each considered feature. In this way, the evolution is charged of evolving the best possible function that extract color, orientation, or shape information; without focusing on the problem of achieving invariance. Thus, the hierarchical structure helps to achieve the desired result through function composition. Next, we describe the EVO features that are used within the artificial visual cortex.

#### 4.1.1 Evolved Color Map

The color image received as input is transformed with a function,  $EVO_C : I_{color} \rightarrow VM_C$ , that enhance the color feature. In this way, an  $EVO_C$  is evolved with genetic programming to optimize the extraction of color information of the objects within the image. The result is an image or visual map  $VM_C$  containing the prominence in color that represents the best feature’s image in color space. Thus, the evolutionary process uses the set of functions and terminals provided in Table 2. The notation is summarized as follows,  $I_{T_C}$  can be any of the terminals in  $T_C$ , as well as the output of any of the functions in  $F_C$ ; the  $Complement(I_{T_C})$  function symbolizes a negative image that is represented by the inversion of an image.

### 4.1.2 Evolved Orientation Map

The function used to compute the orientation,  $EVO_O : I_{color} \rightarrow VM_O$ , is evolved with genetic programming to optimize the extraction of edge information within the input image. The result of this operation is a visual map  $VM_O$  in which the pixel values represent the feature prominence, in such a way, that the higher the pixel value the greater the prominence of the feature. In this way, genetic programming applies the functions and terminals of Table 1 in order to enhance the best orientation features that are useful for the object recognition task. The notation used is as follows.  $I_{T_O}$  can be any of the terminals in  $T_O$ ; as well as, the output of any of the functions in  $F_O$ ;  $D_u$  symbolizes the image derivatives along direction  $u \in \{x, y, xx, yy, xy\}$ ;  $G_\sigma$  are Gaussian smoothing filters with  $\sigma = 1$  or 2.

**Table 2** Set of functions and terminals used by  $EVO_O$  to create the visual map  $VM_O$

$F_O = \{+, -, \times, \div,  +,  -, \sqrt{I_{T_O}}, I_{T_O}^2, \log_2(I_{T_O}),$ $G_{\sigma=1}, G_{\sigma=2},  I_{T_O} , \frac{I_{T_O}}{2}, D_x, D_y\}$
$T_O = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, G_{\sigma=1}(I_r),$ $G_{\sigma=2}(I_r), D_x(I_r), D_y(I_r), D_{xx}(I_r), D_{yy}(I_r),$ $D_{xy}(I_r), \dots\}$

### 4.1.3 Evolved Shape Map

The function used to compute the shape features,  $EVO_S : I_{color} \rightarrow VM_S$ , is evolved with genetic programming to optimize the extraction of shape information in the input image. The result of this operation is a visual map  $VM_S$  that provides the form and structure of the object of interest within the image. In this way, genetic programming applies the functions and terminals of Table 3. We would like to remark that the application of this kind of morphological functions has not been applied in previous research regarding the ventral and dorsal streams. Thus, according to

**Table 3** Set of functions and terminals used by  $EVO_S$  to create the visual map  $VM_S$

$F_S = \{+, -, \times, \div, \text{round}(I_{T_S}), \text{floor}(I_{T_S}), \text{ceil}(I_{T_S}),$ $\text{dilation}_{\text{diamond}}(I_{T_S}), \text{dilation}_{\text{square}}(I_{T_S}),$ $\text{dilation}_{\text{disk}}(I_{T_S}), \text{erosion}_{\text{diamond}}(I_{T_S}),$ $\text{erosion}_{\text{square}}(I_{T_S}), \text{erosion}_{\text{disk}}(I_{T_S}), \text{skeleton}(I_{T_S}),$ $\text{boundary}(I_{T_S}), \text{hit} - \text{miss}_{\text{diamond}}(I_{T_S}),$ $\text{hit} - \text{miss}_{\text{square}}(I_{T_S}), \text{hit} - \text{miss}_{\text{disk}}(I_{T_S}),$ $\text{top} - \text{hat}(I_{T_S}), \text{bottom} - \text{hat}(I_{T_S}), \text{open}(I_{T_S}),$ $\text{close}(I_{T_S})\}$
$T_S = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$

the literature the work reported in this paper could be considered as the first to use morphological image processing within the modeling of the visual cortex.

Finally, in order to obtain the intensity of an input image  $I_{color}$ , we apply a similar process described in previous research where the red, green, and blue values of each pixel are averaged. The formula is developed as a function  $VM_I : I_{color} \rightarrow I$ , that is obtained with the following formulae  $VM_I = \frac{I_r + I_g + I_b}{3}$ .

#### 4.1.4 Conspicuity Maps

The conspicuity maps ( $CMs$ ) are obtained by means of a center-surround function, which is applied to the visual maps in order to simulate a set of center-surround receptive fields. This natural structure allows the ganglion cells to measure the differences between firing rates in center ( $c$ ) and surroundings ( $s$ ) of ganglion cells. First, a pyramid  $VM_I(\alpha)$  of nine spatial scales  $S = \{1, 2, \dots, 9\}$  is created for each of the four resulting  $VMs$ . Afterwards, an across-scale subtraction  $\ominus$  is performed, resulting in a center-surround map  $VM_I(\omega)$  in such a way that the value of the pixel is augmented as long as the contrast is increased within their neighbors at different scales. Finally, the  $VM_I(\omega)$  maps are added using an across-scale addition  $\oplus$  in order to obtain the desired conspicuity maps  $CM_I$ .

Until this stage, we have four  $CMs$ , one for each feature, as shown in Figure 2. The  $CMs$  are obtained similar to Walther and Koch model [24]. Next, instead of combining the  $CMs$  into a single saliency map, the idea here is to use the four  $CMs$  as input to an artificial ventral stream in order to derive a vector descriptor, which will be used by a classifier. In fact, the fitness function is computed from the accuracy achieved with a support vector machine (SVM).

## 4.2 Artificial Ventral Stream (AVS)

Now, that all regions have been highlighted; the next step is to describe such important regions. The typical approach is based on a template matching technique between the information obtained with an interest region selection process and a number of prototype patches. Traditionally, the goal is to learn a set of prototypes that are known as the universal dictionary of features and which are used to identify all object categories. Hopefully, the SVM can recognize the prototypes that correspond to a specific image of a given category. On the other hand, the proposal in this paper is to optimize the functionality of the ventral stream that is evolved with the aim of enhancing the set of prominent features that were highlighted during the interest region detection computed in previous stages. Thus, in this work the selection of interest regions is performed by the artificial dorsal stream through the transformation of the conspicuity maps. It should be noted that according to the artificial ventral stream each evolved function is a composite function that is

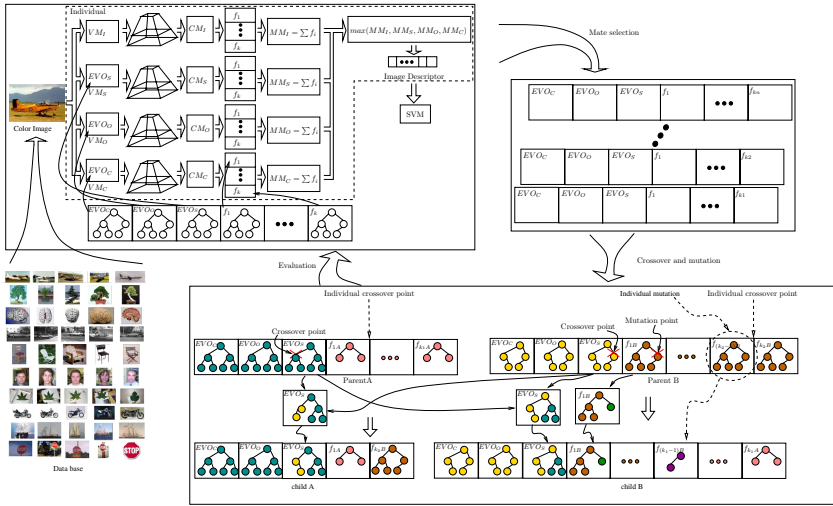
capable of substituting several prototype features; thus, reducing significantly the total number of operations needed to define all object features that are used to describe and classify the input images. According to Figure 2 the information provided by the conspicuity maps is feedforward to  $k$  operators that emulate a set of lower order hypercomplex cells replicating the functionality of a virtual tissue. Thus, all evolved functions along each dimension are added in order to obtain a single measure that we called mental map. Hence, a mental map is obtained for each dimension: color, orientation, shape, and intensity. In this way, all mental maps are combined with a max operation that is used to highlight the necessary characteristics that recognize a specific object class. Note, that each function is an evolved visual operator (EVO) built by several GPs from the particular set of terminals and functions shown in Table 4. Note also that this second stage could be said to perform an information description operation (IDO) with the aim of discovering the best set of functions that creates the most discriminant vector of characteristics. Hence, this set of functions replaces the universal dictionary proposed in [27, 18, 24] and we claim that it corresponds to a function driven approach.

**Table 4** Set of functions and terminals for the ventral stream

Functions:	$+$ , $-$ , $/$ , $*$ , $ - $ , $ + $ , $\sqrt{\cdot}$ , $(\cdot)^2$ , $\log(\cdot)$ , $D_x(\cdot)$ , $D_y(\cdot)$ , $D_{xx}(\cdot)$ , $D_{xy}(\cdot)$ , $D_{yy}(\cdot)$ , $Gauss_{\sigma_1}(\cdot)$ , $Gauss_{\sigma_2}(\cdot)$ , $0.05(\cdot)$
Terminals:	$C1$ , $D_x(C1)$ , $D_{xx}(C1)$ , $D_y(C1)$ , $D_{yy}(C1)$ , $D_{xy}(C1)$

## 5 Evolving AVCs with Organic Genetic Programming

This section describes the main aspects for the evolution of AVCs through the application of what we called organic genetic programming (OGP). All elements introduced in the OGP embody an organic motivation, in a sense of describing an organ composed of tissues, which could be part of an artificial living organism. Figure 3 illustrates the complexity of the proposed system using a kind of heterogeneous and hierarchical genetic programming. In our model the genotype is built from several trees that can be seen as the genes and which are arranged into a complex chromosome. The phenotype is decoded according to Figure 2. Thus, the algorithmic process that mimics the visual information processing of an AVC should be seen as a single entity. In other words, the functional representation of the artificial organ is represented by the whole hierarchical and heterogeneous structure. The representation that is proposed has the aim of ensuring the development of complex functions, while freely increasing the number of programs according to the task at hand; in this case, the classification of several object classes. In this way, the structure can grow in the number and size of its elements. Hence, it is important to note that each individual

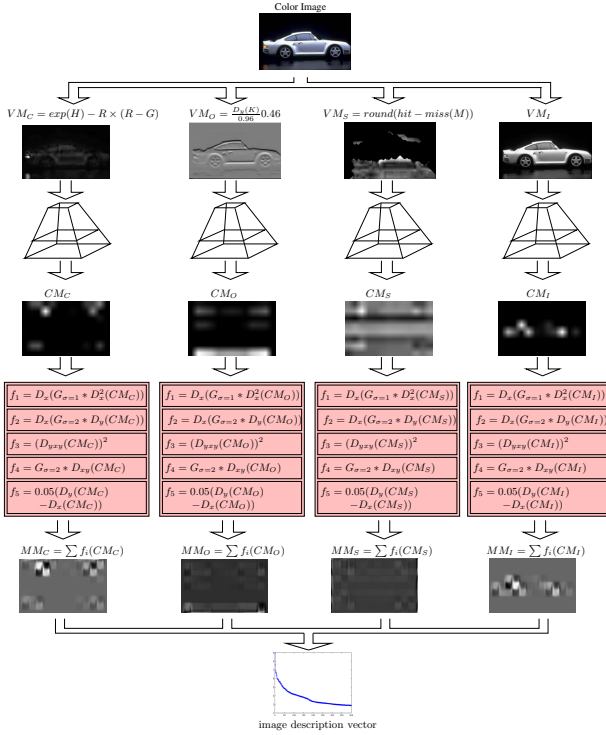


**Fig. 3** General flowchart of the methodology to synthesize an artificial visual cortex

within a population should be understood as the whole AVC and it is therefore not only a list of tree-based programs, but the whole information processing depicted in Figure 2

## 6 Experimental Results

This section provides details about the experiments in order to explain the system that was implemented to learn an artificial visual cortex. All experiments were performed in a Dell Precision T7500 Workstation, Intel Xeon 8 Core, NVIDIA Quadro FX 3800 and Linux OpenSUSE 11.1 operating system. The system was tested using 10 classes and 15 images per class of the Caltech 101 database, see [7]. The classifier used in the experiments was the SVM implementation developed by Chan and Lin, see [3], in order to compare with the HMAX model [24]. Table 5 presents a summary of the best results and a comparison with the HMAX model, an implementation HMAX-CUDA, and a previous proposal called the artificial ventral stream (AVS), see [4]. Note, that the total number of convolutions is much lower than the HMAX and HMAX-CUDA. This aspect is important since the factor of improvement is on the order of hundred of operations. However, the performance of the AVC is lower than the HMAX model but its level is worse in testing, while the effectiveness of our approach remains constant. Figure 4 shows the run where the best program was obtained, see Figure 4. Also, the Figure 6 illustrates the range of



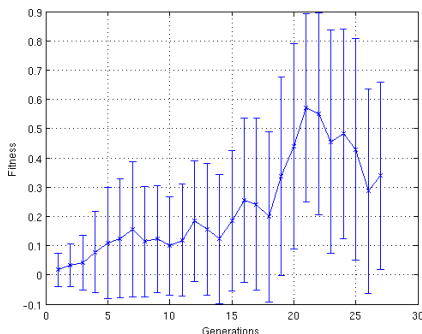
**Fig. 4** Flowchart of the best individual achieved with the methodology to synthesize an artificial visual cortex

descriptor values of the best solution for each class. We provide also the overall results of the best AVC through the confusion matrix, see Table 7. Due to the level of accuracy being achieved by the AVC we decide to make a simpler test. We evaluated the performance of the proposed model in the object present/absent experiment using several object classes from the same CalTech data set. In this experiment, each data set was randomly divided in two sets for training and testing using 50 images for each set out of 800 images. We remark that for this case the algorithm scores a perfect solution during the initial random population. Therefore, it was not necessary to evolve the AVC to find a solution to the problem. Table 6 shows a comparison with the HMAX model using boost and SVM as classifiers for the following classes: airplanes, cars, faces, leaves, and motorbikes.

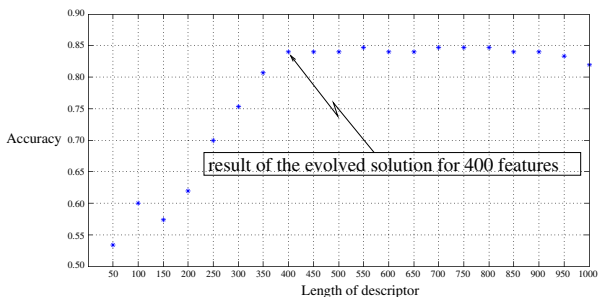


**Table 5** This table shows the comparison of performances between HMAX, HMAX-CUDA, AVS and AVC

	Image size	HMAX MATLAB	HMAX CUDA	Artificial V. S.	Artificial V. C.
Running time	896 × 592	34s	3.5s	2.6s	9.91s
over different image size	601 × 401	24s	2.7s	1.25s	5.32s
	180 × 113	9s	1s	0.23s	0.49s
Performance over 15 training images per 10 classes		94%	94%	78%	85.3%
Performance over 15 testing images per 10 classes		73%	73%	80%	84%
Number of convolutions		4848	4848	216	95



(a) Average fitness with standard deviation



(b) Behavior of accuracy with respect to the descriptor length

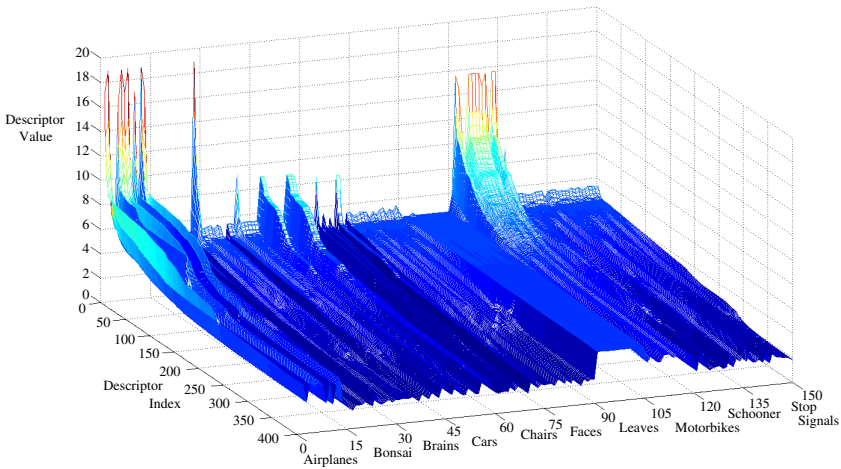
**Fig. 5** Figure (a) shows the average fitness and standard deviation of the run that produces the best individual. Figure (b) depicts the performance after changing the descriptor length.

**Table 6** This table shows the performance comparison between HMAX, HMAX-CUDA, and AVC. Note, that in the case of the HMAX model a learning process was necessary to identify the best patches, while for the AVC only a random sampling to discover the best solution.

Data sets	Performance of HMAX		Artificial V. C.
	boost	SVM	
Airplanes	96.7 %	94.9 %	100 %
Cars	95.1 %	93.3 %	100 %
Faces	98.2 %	98.1 %	100 %
Leaves	97.0 %	95.9 %	100 %
Motorbikes	98.0 %	97.4 %	100 %

**Table 7** This table shows the results of the best solution in the form of a confusion matrix obtained during the AVC testing. The final accuracy  $acc = 84\%$  classifies correctly (126/150) images.

	Airplanes	Bonsai	Brains	Cars	Chairs	Faces	Leaves	Motorcycle	Schooner	Stop Signal
Airplanes	15	0	0	0	0	0	0	0	0	0
Bonsai	0	9	2	0	3	0	0	0	0	1
Brains	0	2	11	0	1	0	0	0	1	0
Cars	0	0	0	14	0	0	0	0	1	0
Chairs	0	0	2	1	11	0	0	0	0	1
Faces	0	0	0	0	0	14	0	0	1	0
Leaves	0	0	0	0	0	0	15	0	0	0
Motorcycle	0	0	0	0	0	0	0	15	0	0
Schooner	0	1	0	0	0	0	0	0	12	2
Stop Signal	0	2	1	0	0	1	0	0	1	10



**Fig. 6** This plot shows the descriptors of the best individual that are used as input to the SVM.

## 7 Conclusions

This work shows that a complex program, mimicking an artificial visual cortex, with numerous trees can be evolved to approach successfully a multi-object recognition problem.

**Acknowledgements.** This research was funded by CONACyT through the Project 155045 - “Evolución de Cerebros Artificiales en Visión por Computadora”. First author supported by scholarship 164884 from CONACyT. This research was also supported by TESE through the project DIMI-MCIM-004/08.

## References

1. Ayala, F.J.: Teleological explanations in evolutionary biology. *Philosophy of Science* 37(1), 1–15 (1970)
2. Barton, R.A.: Visual specialization and brain evolution in primates. *Proceedings of the Royal Society of London Series B-Biological Sciences* 265(1409), 1933–1937 (1998)
3. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011), Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
4. Clemente, E., Olague, G., Dozal, L., Mancilla, M.: Object Recognition with an Optimized Ventral Stream Model Using Genetic Programming. In: Di Chio, C., Agapitos, A., Cagnoni, S., Cotta, C., de Vega, F.F., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Langdon, W.B., Merelo-Guervós, J.J., Preuss, M., Richter, H., Silva, S., Simões, A., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Togelius, J., Urquhart, N., Uyar, A.Ş., Yannakakis, G.N. (eds.) *EvoApplications 2012*. LNCS, vol. 7248, pp. 315–325. Springer, Heidelberg (2012)
5. Creem, S.H., Proffitt, D.R.: Defining the cortical visual systems: “what”, “where”, and “how”. *Acta Psychologica* 107, 43–68 (2001)
6. Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. *Annu. Rev. Neurosci.*, 193–222
7. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, p. 178 (2004)
8. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36, 193–202 (1980)
9. Hubel, D.H.: Exploration of the primary visual cortex. *Nature*, 515–524 (1982)
10. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurones in the cat’s striate cortex. *J. Physiol.* 148(3), 574–591 (1953)
11. Itti, L., Koch, C.: Computational modelling of visual attention. *Nature Review Neuroscience* 2(3), 194–203 (2001)
12. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. *Hum. Neurobiol.* 4(4), 219–227 (1985)
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324 (1998)
14. Lennox, J.G.: Darwin was a teleologist. *Biology and Philosophy* 8(4), 409–421 (1993)
15. Mel, B.W.: Seemore: Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation* 9, 777–804 (1997)
16. Milanese, R.: Detecting salient regions in an image: from biological evidence to computer implementation. PhD thesis, Department of Computer Science, University of Genova, Switzerland (December 1993)
17. Milner, A.D., Goodale, M.A.: *The Visual Brain in Action*, 2nd edn. Oxford University Press, Oxford (2006)
18. Mutch, J., Lowe, D.G.: Object class recognition and localization using sparse features with limited receptive fields. *Int. J. Comput. Vision* 80, 45–57 (2008)
19. Oram, M.W., Perrett, D.I.: Modeling visual recognition from neurobiological constraints. *Neural Networks* 7(6), 945–972 (1994)
20. Rensink, R.A.: The dynamic representation of scenes. *Visual Cognition* 7(1-3), 17–42 (2000)

21. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. *Nature Neuroscience* (11), 1019–1025
22. Schneider, G.E.: Contrasting visuomotor functions of tectum and cortex in the golden hamster. *Psychologische Forschung* 31(1), 52–62
23. Schneider, G.E.: Two visual systems. *Science* 163(3870), 895–902 (1969)
24. Serre, T., Kouh, C., Cadieu, M., Knoblich, G., Kreiman, U., Poggio, T.: A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Technical report, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, CBCL-259 (2005)
25. Short, T.: Darwin’s concept of final cause: neither new nor trivial. *Biology and Philosophy* 17, 323–340 (2002), doi:10.1023/A:1020173708395
26. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. *Cognitive Psychology* 12(1), 97–136 (1980)
27. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermediate complexity and their use in classification. *Nature Neuroscience* 5(7), 682–687 (2002)
28. Ungerleider, L.G., Haxby, J.V.: “What” and “where” in the human brain. *Current Opinion in Neurobiology* 4(2), 157–165 (1994)
29. Ungerleider, L.G., Mishkin, M.: Two Cortical Visual Systems, ch. 18, pp. 549–586 (1982)
30. Walther, D., Koch, C.: Modeling attention to salient proto-objects. *Neural Networks* 19(9), 1395–1407 (2006)
31. Wolfe, J.M.: Visual Attention, 2nd edn., ch. 8, pp. 335–386. Academic Press (2000)

# Evolving an Artificial Dorsal Stream on Purpose for Visual Attention

León Dozal, Gustavo Olague\*, and Eddie Clemente

**Abstract.** Visual attention is a natural process performed by the brain, whose functionality is to perceive salient visual features, and which is necessary since it is impossible to focus your sight at two things during the same indivisible time. This work is devoted to the task of evolving visual attention programs through organic genetic programming. The idea is to state the problem of visual attention, which is normally divided in two parts: bottom-up and top-down, in terms of a unique approach based on a teleological framework. Indeed, this paper explains how visual attention could be understood as a single mechanism that is designed according to a given purpose. In this way, genetic programming is used to design top-notch visual attention programs. Experimental results show that this new approach can contrive solutions useful in the solution of “top-down and bottom-up” visual attention problems. In particular, we present a solution to the size popout problem that was unsolved previously in the literature.

## 1 Introduction

The brain can be extremely complex and despite rapid scientific progress much about how the brain works remains a mystery. In nature, there is a large diversity of brain anatomies that are characterized by the specialization of visual systems. Such diversity shows the power of evolution through adaptation. In this way, it has been argued that the evolution of specific visual mechanisms in the primate brain is the

---

León Dozal · Gustavo Olague

CICESE, Carretera Ensenada-Tijuana No.3918, Zona Playitas, Ensenada, B.C., México

e-mail: [leon.dozal@gmail.com](mailto:leon.dozal@gmail.com), [olague@cicese.mx](mailto:olague@cicese.mx)

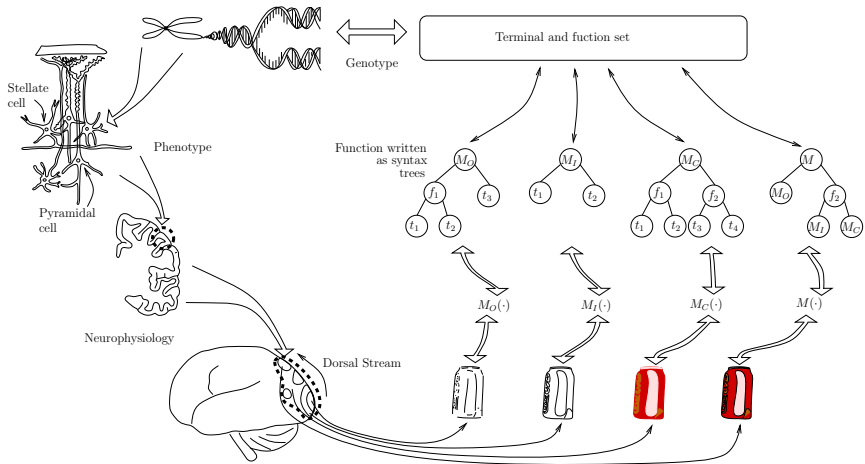
Eddie Clemente

Tecnológico de Estudios Superiores de Ecatepec, Avenida Tecnológico S/N,

Esq. Av. Carlos Hank González, Valle de Anáhuac, Ecatepec de Morelos

e-mail: [eddie.clemente@gmail.com](mailto:eddie.clemente@gmail.com)

\* Corresponding author.



**Fig. 1** This figure illustrates the analogy between the natural and artificial systems. The idea is based on replicating the functionality of a set of artificial tissues that conform what we called the organic genetic programming (OGP).

product of natural selection [2]. Contrary, in the past it was widely believed that human observers construct a complete representation of everything in their visual field [6, 21]. This has been amply refuted by a large amount of research. The visual attention is without a doubt one of the most important mechanisms in the visual system because the brain, or visual cortical areas, are unable to process all information received along the entire visual field. In this way, there are two basic process that define the problem of visual attention. The first basic phenomenon is the limited capacity of information processing. At one given time, only a small amount of information available to the retina can be processed and used for control of some specific behavior. The second basic phenomenon is selectivity; in other words, the ability to filter unwanted information [5].

In this work, we follow the idea that visual attention is controlled by both cognitive, or top-down (TD) factors, such as knowledge, expectation, and current goals; as well as, reactive stimulus, or bottom-up (BU) factors, that refers to sensory stimulation like gaze, focus, and cuing, see [4]. Moreover, the low level mechanisms for feature extraction act in parallel over the entire visual field using the TD and BU systems in order to provide the signs that highlight the image regions. Afterwards, attention is focused sequentially on the highlighted regions of the image in order to analyze them in more detail [22, 12].

## 1.1 Problem Statement

As was mentioned previously selectivity is a quality of the visual attention process. Today, many researchers believe that it is necessary to implement this property

within artificial systems. The answers to the questions: what features should it use? and when to use those features? is not evident. Moreover, a problem arise after the feature detection stage known as feature combination. Combining different features, such as color, orientation and shape, within a single saliency representation becomes complex since these features came from different visual dimensions. The complexity increases when you are looking for a particular object and it is necessary to filter the information to stress the features of the desired object. In this work organic genetic programming (OGP) is used to address this problem. In this way, the idea is to apply OGP as the mechanism to obtain the most suitable visual attention programs (VAPs) that are capable of pursuing different goals.

## 2 Visual Attention Processing

This section proposes a new approach for visual attention with the aim of organizing the whole system as a single functional entity that changes its operation according to a purpose, but without changing its structure. In this way, contrary to most traditional approaches that represent the visual attention model through the division of the process into reactive and volitive parts, our proposal provides the simplicity and uniqueness to endow a machine with the ability of designing visual attention programs. Next, the main works are reviewed in order to understand our definition of visual attention.

In the early 80's the feature-integration theory for visual attention was proposed by Treisman and Gelade[22]. Actually, this theory is considered as the most widely accepted paradigm for visual attention within the cognitive sciences community, and it is used as the fundamental computational model for “bottom-up” visual attention. In a first stage, the feature-integration theory proposes that the features of the whole visual scene are perceived in parallel. Next, in a second stage, all features are detected through the stimuli and are integrated into a coherent representation. In a third stage, the stimuli are processed serially by focusing the visual attention on them. In this way, when visual attention is fixed on a particular stimulus, the characteristics, around the attended area, are merged to form a single object. Therefore, it is said that visual attention should serve as a “glue” that combines the features of an individual object to obtain a unique and coherent representation.

### 2.1 Classical Approach to Visual Attention

The visual attention functionality, regarding the localization of objects, is related to the brain areas around the dorsal stream. Thus, the dorsal pathway is defined as projecting from V1 through V2, V3, middle temporal area (MT), medial superior temporal area (MST) and finally to the posterior parietal cortex, see [23]. Nevertheless, there is a lack of consensus about the specific brain areas, structure and functionality, that conform the dorsal stream. For example, in another theory the dorsal stream is also known as the “how” stream [15]; while, in the work described in [1] the dorsal stream areas do not correspond to the literature.

Nowadays, classical explanations of visual attention are in agreement that the dorsal stream could be seen as a theory that is influenced by BU and TD factors. In this way, it is affirmed by [4] that there are two interacting neural systems involved in the control of BU and TD factors that control visual attention. As a result, the dichotomy of visual attention has inspired several computational models that are commonly based on only one of these two factors. For example, the research in computational neuroscience has traditionally separated their study; as well as, the implementation of visual attention using a benchmark system for human-visual gaze estimation [17, 3] or for the solution of object recognition tasks. Contrary to this line of research, we propose to study visual attention from a teleological standpoint as a way of unifying through this framework both factors with the intention of considering visual attention as a single mechanism.

Next, both BU and TD factors are reviewed in order to introduce our approach with the aim of understanding the structure and functionality of visual attention. In this way, both factors should be studied through a unique process that is capable of adapting itself according to the pursued goal or goals.

### 2.1.1 “Bottom-Up” Control for Visual Attention

In the literature the idea of BU visual attention is related with involuntary attention, which is usually compared with the concept of a spotlight. This metaphor has been used by Posner *et al.* [18] to explain that visual attention operates “as a spotlight which improves the detection of events in their proximity”.

Actually, one of the best and easiest ways of implementing a set of tests is to study BU attention in terms of visual search. Commonly, the exploratory task is studied experimentally using a set of images containing challenging visual stimuli that are presented to an observer. For each image there is an object called target that is different from the rest. Today, the existing computational models are mostly bottom-up models based on the feature-integration theory [22]. The first biologically, neurologically, and plausible computational model for BU visual attention was proposed by Koch and Ullman [12]. Later, Milanese [14] proposed a visual attention system that uses mechanisms, inspired from biological processes, which were adopted by the research community to create a whole new trend in visual attention systems. Some of these processes are the color opponencies such as: red-green and blue-yellow; as well as, the center-surround difference present in the receptive field of the cortical cells. One of the most well-known models is probably that of Itti *et al.* [10], which provided a software that popularize these theoretical processes. In summary, this can be considered as a very detailed model that proposes simple solutions to complex issues. Thereafter, another breakthrough was proposed by Rensink [20, 19] who introduced the notion of proto-objects and the interpretation of the apparent blindness of observers to recognize dramatic changes within a scene. Finally, Walter and Koch [24], showed that the proposed model can enhance the task of object recognition through the application of the concept of proto-object for visual attention tasks.



### 2.1.2 “Top-Down” Control for Visual Attention

Today, there is an agreement that TD cues play a key role in the processing of visual information. In particular, it is known that there are numerous connections between higher and simpler information processing areas. In this way, it is said that voluntary attention takes more time and effort to accomplish compared to involuntary attention. This is because the target shares with the distractors two or more features, which forces the observer to perform a scanning of the whole scene.

The TD visual phenomenon, just explained, is studied in psychophysics; in fact, TD factors are usually investigated through the so-called “cuing experiments”. This type of experiments consists in presenting a “cue” that guides the observer’s attention toward the target. In this way, it is said that cues may indicate *where* is the target, like in the case of an arrow pointing towards the target, or by answering the question of *what* is the target by means of finding the similarities between a picture, or written description of the target, see [7]. Thus, there have been several attempts to implement models using TD cues. For example, Oliva *et al.* propose an attentional model that uses knowledge about the distribution of features over the image in order to select salient regions [16]. Peters and Itti [17] proposed a combined model BU/TD, in which they measure the ability of the model to predict the saccades of people playing video games. In this way, they improved the prediction by a margin that doubles the performance obtained by the BU model. The TD part computes a feature vector describing the “gist” of the image with the positions of saccades obtained from real observers that are used to train the model. Finally, a feature vector is calculated to generate the saccades prediction map. Recently, Borji, *et al.* [3] follow the same line of research proposed by Peters and Itti, but the system is based on a different approach that determines the position of the saccades with respect to the observer by applying a set of robust classifiers.

Thus, from a computational modeling standpoint TD factors are not a trivial task; in other words, emotions and desires are difficult concepts to model; hence, such kind of cognitive concepts are still challenging within computer science. Nevertheless, from a pragmatic point of view, there are goals to achieve. Therefore, a purpose should not be confused with a desire; when we refer to a purpose, we talk in terms of whether the goals are achieved or not. Here lies the importance of modeling TD and BU mechanisms in teleological terms.

## 2.2 An Unified Approach of Visual Attention

Aristotle defined the final cause or *telos* as that for which something is done, its purpose. Also distinguishes between the *telos* and desire, consciousness and intelligence. Therefore, according to Aristotle, an organism like a seed has a purpose just as a person. Latter, Kant [11] wrote, in the “Analytic of Teleological Judgment”, that organisms must be regarded in teleological terms, and in the “Dialectic of Teleological Judgment”, he attempts to reconcile this teleological conception of organisms with a mechanistic account of nature. Everything can be completely explained by causality, except the organisms.

From our standpoint, attention is the result of a single mechanism that is designed to obey a general purpose; as well as, different particular purposes. For example, the most primitive purpose for life could be survivorship. But, the achievement of survivorship depends on many other particular purposes; for example, prey hunting, mating, predator escape, etc. In this sense, visual attention is capable to adapt itself to the kind of goal that depend on the current purpose of the organism. In order to accomplish such task, it is necessary a unique and general visual attention structure capable of performing, by some temporal readaptation, the necessary functions to achieve that aim or intention. Furthermore, considering the fact that most of the tasks involved in the design of BU and TD factors are complex, we could say that the space of possible readaptations is at least very large and discrete. Therefore, we defined visual attention as follows.

**Definition 1 (Visual Attention)** *Visual attention is a process that designs a relationship between the different properties of the scene, which are perceived through the visual system with the aim of selecting a particular aspect.*

For these reasons, we consider the visual attention as a single computational structure that performs BU and TD processes. In consequence, in this work visual attention is studied within a unified framework in order to evolve visual attention programs (VAPs) that will be adapted for specific purposes.

### 3 Purposive Evolution for Visual Attention

The theory of evolution is not exempt of the concept of purpose. Charles Darwin was the one who brings the concept of purpose into consideration. Note that Darwin uses the term final cause systematically in his writings as it is documented by Lennox [13]. On the other hand, Barton [2] explains the evolution of primates brains in terms of the specialization of visual mechanisms; such as visual attention. Thus, this section describes the general structure of attention, which is biologically inspired and will be evolved to suit different purposes. The resulting evolved programs will be known as visual attention programs (VAPs). Moreover, following the same direction of Treisman, the description of the general approach is divided into two main stages: acquisition and integration.

#### 3.1 Acquisition of Early Visual Features

In previous works of artificial visual attention, the operators are established according to the knowledge in neuroscience. Moreover, it is widely recognized that the operation of the visual cortex, specifically the dorsal stream, is a product of the evolutionary process. In this way, we propose to use evolutionary computation to obtain these artificial visual operators. In summary, this work explains how to use specialized evolved visual operators (EVOs) for the acquisition of visual dimensions such as color, orientation and shape. Next, the EVO features used within the VAP are defined.

**Table 1** Functions and terminals used by  $EVO_O$  to create the orientation visual map  $VM_O$

$F_O = \{+, -, \times, \div,   +  ,   -  , \sqrt{I_{T_O}}, I_{T_O}^2, \log_2(I_{T_O}),$ $G_{\sigma=1}, G_{\sigma=2},  I_{T_O} , \frac{I_{T_O}}{\sigma}, D_x, D_y\}$
$T_O = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, G_{\sigma=1}(I_r), G_{\sigma=2}(I_r),$ $D_x(I_r), D_y(I_r), D_{xx}(I_r), D_{yy}(I_r), D_{xy}(I_r), \dots\}$

### 3.1.1 Orientation

In previous work the characteristic of orientation for images was only computed in gray scales. Thus, our work proposes to evolve the property of orientation along the different color bands of the image. In this way, a rich set of information is generated because the edges, corners, and other similar features could appear more highlighted with the color bands. Therefore, the evolutionary approach evolves a function  $EVO_O : I_{color} \rightarrow VM_O$  that cooperates with the VAP in order to accomplish a purpose. The resulting  $EVO_O$  operation is a visual map  $VM_O$  for which the pixel value represents the feature prominence; in such a way, that the larger the pixel value, the greater the orientation prominence of the feature. This computation is performed through a set of functions and terminals that are provided in Table 1. The notation that was used is as follows.  $I_{T_O}$  can be any of the terminals in  $T_O$ ; as well as, the output of any of the functions in  $F_O$ ;  $D_u$  symbolizes the image derivatives along direction  $u \in \{x, y, xx, yy, xy\}$ ;  $G_\sigma$  are Gaussian smoothing filters with  $\sigma = 1$  or  $2$ .

### 3.1.2 Color

In biology, the color is encoded through photoreceptor cells known as cones, which are located in the retina. However, a special case is the yellow color which is not perceived in the cones but in the retinal ganglion cells. Then, the dorsal pathway is composed of several tissues V1, V2 and V4, whose cells respond to color features. In this work the characteristics of color information that will be used as the building blocks to construct the VAPs are color opponencies and simple arithmetic operations between the different color bands in the corresponding color space. In the same way, as in  $EVO_O$ , the evolutionary process uses a set of functions and terminals provided in Table 2 to evolve the feature in the color space. The result is a visual map  $EVO_C : I_{color} \rightarrow VM_C$  containing the color prominent features.

**Table 2** Functions and terminals used by  $EVO_C$  to create the color visual map  $VM_C$

$F_C = \{+, -, \times, \div,   +  ,   -  , \sqrt{I_{T_C}}, I_{T_C}^2, \log_2(I_{T_C}),$ $Exp(I_{T_C}), Complement(I_{T_C})\}$
$T_C = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, RG_{oppn}, YB_{oppn}\}$

### 3.1.3 Shape

As in previous dimensions, the evolutionary process uses a set of functions and terminals provided in Table 3 to characterize the shape information. Note that we propose to describe these features through mathematical morphology. The result is a visual map  $EVO_S : I_{color} \rightarrow VM_S$  containing the shape prominent features. This part is evolved with genetic programming with the aim to provide the information about shape and structure of the object of interest within the image. We would like to remark that the application of this kind of morphological functions has not been applied in previous research studying the ventral and dorsal streams.

**Table 3** Set of functions and terminals used by  $EVO_S$  to create the shape visual map  $VM_S$

$F_S = \{+, -, \times, \div, \text{round}(I_{T_S}), \lfloor I_{T_S} \rfloor, \lceil I_{T_S} \rceil, \\ \text{dilation}_{diamond}(I_{T_S}), \text{dilation}_{square}(I_{T_S}), \\ \text{dilation}_{disk}(I_{T_S}), \text{erosion}_{diamond}(I_{T_S}), \\ \text{erosion}_{square}(I_{T_S}), \text{erosion}_{disk}(I_{T_S}), \text{skeleton}(I_{T_S}), \\ \text{boundary}(I_{T_S}), \text{hit} - \text{miss}_{diamond}(I_{T_S}), \\ \text{hit} - \text{miss}_{square}(I_{T_S}), \text{hit} - \text{miss}_{disk}(I_{T_S}), \\ \text{top} - \text{hat}(I_{T_S}), \text{bottom} - \text{hat}(I_{T_S}), \text{open}(I_{T_S}), \\ \text{close}(I_{T_S}) \}$
$T_S = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$

Finally, to obtain the intensity of pixels in the image the model averages the red, green and blue values for each pixel. The result of this operation is a visual map  $VM_I$  in which the pixel represents the prominence over the intensity space.

### 3.1.4 Computing the Conspicuity Maps

The conspicuity maps ( $CMs$ ) are obtained by means of a center-surround function that is applied in order to simulate the center-surround receptive fields. This natural structure allows the ganglion cells to measure the differences between firing rates in center ( $c$ ) and surroundings ( $s$ ) areas of ganglion cells. At this stage, we have one  $CM$  for each feature. The  $CM$  is obtained as proposed in the Walther and Koch model [24]. Finally, the  $CMs$  are combined to obtain a single saliency map as explained in the next section.

## 3.2 Feature-Integration for Visual Attention

The saliency map ( $SM$ ) defines the place for the most prominent locations of the image; given the characteristics of intensity, orientation, color and shape. In other words, the objective of this stage is to decide where attention could be directed at any given time. In this work, the problem statement considers that the task must be

**Table 4** Set of functions and terminals used by *EFI* to create the object saliency map *OSM*

$F_{fi} = \{+, -, \times, \div,   +  ,   -  , \sqrt{I_{fi}}, I_{fi}^2, Exp(I_{fi}),$ $G_{\sigma=1}, G_{\sigma=2},  I_{fi} , D_x, D_y\}$
$T_{fi} = \{CM_I, CM_O, CM_C, D_x(CM_I), D_y(CM_I),$ $D_{xx}(CM_I), D_{yy}(CM_I), D_{xy}(CM_I), \dots\}$

addressed to achieve a specific goal. As a result, if the task needs to accomplish a purpose; then, the main criterion should be the one that guides the suitable combination of characteristics. Therefore, we decided to evolve the integration of *CMs* through a function that we called Evolved Feature Integration (*EFI*). Therefore, the *VAPs* provide a dynamic structure since the *EVOs* can be selected using a fusion process executed by the *EFI*. This process considers different combinations of *CMs* to complete the entire process. Once the integration of features is performed, we get an optimized saliency map (*OSM*) indicating the location of the most prominent regions within the original image, known as proto-objects (*P<sub>l</sub>*). The definition of the *EFI* function is as follows:

$$EFI : CM_l \rightarrow OSM ; l \in \{O, C, I\}$$

The evolutionary method uses the set of functions and terminals, listed in Table 4, to create a fusion operator that highlights the features of the object of interest.

## 4 Organic Genetic Programming

In this section, we describe the main aspects for the evolution of *VAPs* using the organic genetic programming (OGP) strategy. In the OGP the chromosome is composed of several genes that are represented each one with a tree structure. At the gene level the genetic operations are performed like in the classical genetic programming. While at the chromosome level the whole genotype is described by the parallel set of functions acting over the color, shape, and orientation dimensions. The design of the OGP embody an organic motivation in a sense of describing an organ or tissue, as a part of a living organism, and their complexity. We introduce a set of new concepts in order to deal with the evolution of complex structures, which are explained below.

In the experiments the OGP goal is to discover a program that learns to attend a prominent object using a set of training images. In this work, the *VAP's* genotype is considered as robust because it is capable of encoding the phenotype of an artificial dorsal stream. In other words a genotype consisting of three to four trees is composed of different and specialized operations. Thus, each tree has its own independent set of functions and terminals, which are listed in Tables 1, 2, 3 and 4; according to orientation, color, shape and feature integration respectively.

The first one encodes the orientation feature similar to the orientation-sensitive cells of V1 [9]. The second one represents the color feature in an analogy to the photo-receptor cells presented in the retina; as well as, the color sensitive cells of the layers V1 and V4 of the visual cortex. The third one models the shape feature that characterizes shape-sensitive cells present in layers V2 and V4 of the brain. Finally, the fourth one encodes the way in which the features are combined to obtain the saliency map, or operation of the posterior parietal cortex [8].

## 5 Experiments and Results

The following experiments are divided in two parts, according to the goal that the VAP is attempting to reach. The OGP is basically the same for both experiments, the only parts that change are the fitness function, which encodes the purpose, and the set of images utilized for training. The fitness function of the OGP is the characterization of the purpose, the answer to the “what are the individuals for?”. In other words, it is the way in which the purpose is implemented as a computer programming.

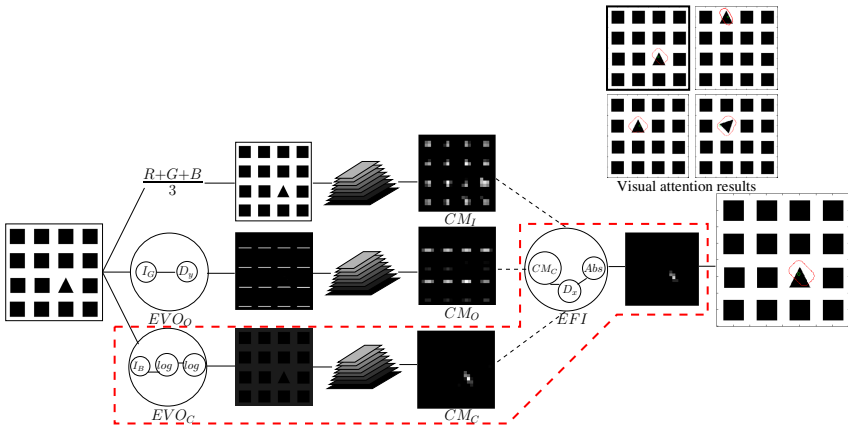
### 5.1 Evolution of VAPs for Aiming Scene Novelty

The first set of experiments are stated in terms of *visual search*, which is commonly applied like in classical research devoted to visual attention. In this way, the tests are designed to obtain through artificial evolution a VAP that is specially adapted to find the novelty, or asymmetries, in a simple set of images of the kind that are used in psychophysical studies.

#### 5.1.1 Search of Appearance Novelty

The first experiment was conceived with the aim of obtaining a VAP capable of centering attention with respect to appearance novelty. In other words, capable of focusing an object using the shape and the information around the object. Figure 2 shows the  $VAP_{triangle}$  that was obtained by the OGP. We can remark that the  $VAP_{triangle}$  utilizes only the color dimension. The proposed solution is to regularize the image through the logarithm function. This process reduces the contrast between the black and white areas, and as a result, the regions around the triangle are highlighted after the central-surround processing and evolved feature integration steps. Thus, the functions obtained by the OGP are listed below:

$$\begin{aligned} EVO_O &:= D_y(I_G) \\ EVO_C &:= \log(\log(I_B)) \\ EFI &:= \|D_x(CM_C)\| \end{aligned}$$



**Fig. 2** Bottom-Up image testing of novelty. This figure depicts the best visual attention program that was evolved with organic genetic programming to attend the triangle.

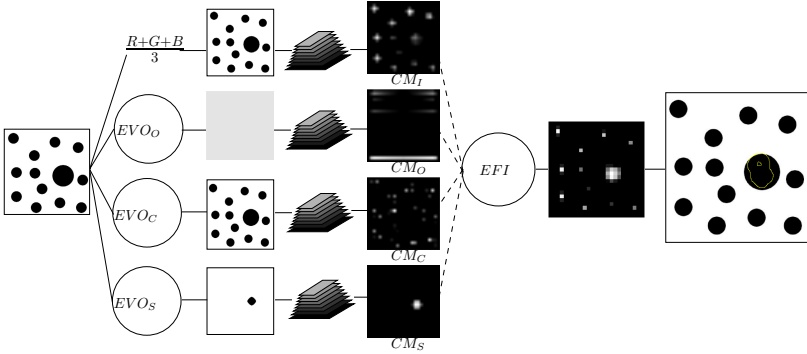
The evolution used only one image per training, This image shows the highlighted black square at the top-right corner of Figure 2. The extra images illustrate the results achieved during a set of tests considering rotation and translation; indeed, the triangle was rightly focused.

**5.1.2 Search of Size Novelty**

The experiment described next is important because, according to the literature, it has not been solved previously by any computational method devoted to the solution of visual attention. The main reason may be due to the overlook of the characteristic of size, and the lack of a suitable choice of functions within the problem statement. Thus, in order to solve this problem for this particular experiment, see Figure 3, it was decided to increase as an extra dimension the property of shape ( $EVO_S$ ), which is computed as explained in section 3.1.3 using the fundamental operations of mathematical morphology. Hence, the best set of functions obtained by the OGP are listed below:

$$\begin{aligned}
 EVO_O &:= D_x D_y D_{xx}(Y) + tresh\left(\frac{D_x D_y D_{xx}(Y) + 0.93}{0.83}\right) \\
 EVO_C &:= I_V - I_R \\
 EVO_S &:= ((I_G + 0.90) \oplus Sqr) \oplus Sqr \\
 EFI &:= \left| D_y(CM_S) - \frac{CM_C}{|CM_O - \frac{CM_C}{D_y(CM_O)}|} \right|
 \end{aligned}$$

where  $Sqr$  denotes a square structuring element over the dilation operator  $\oplus$ .



**Fig. 3** Bottom-Up testing for the size popout problem. This figure provides an example of a pop-out effect, the big circle, that previous visual attention programs were unable to detect, see [7]. Indeed, the evolved visual attention program was able to detect the saliency in the image related to the single big object.

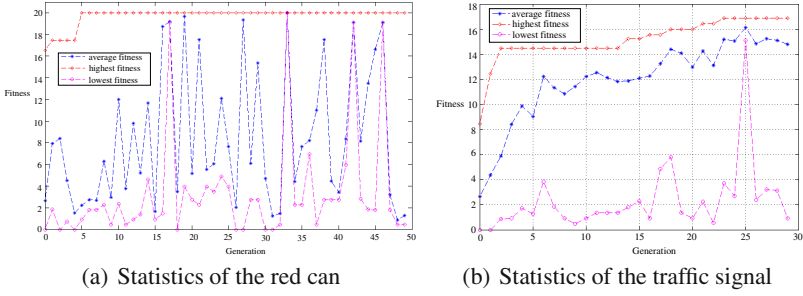
## 5.2 Evolution of VAPs for Aiming Specific Targets

In this section, the obtained VAPs and their performance are presented for the case of TD tasks. Figure 4 provides the statistics of the top-down runs plotting the average, highest, and lowest values. The examples illustrate that for each target object: red can and traffic signals, a solution could be attained without changing the proposed computational framework. During the training stage, the  $FOA_{coke}$  detected the object of interest, in this case the coke, with a successful rate of 100% after using 44 images. In this way, during the testing stage the  $FOA_{coke}$  detected the object of interest with a rate of 88.13% using 59 images. In this way, from the 59 test images the coke was detected in 52 occasions. Moreover, the percentage of detection increases after considering a second attempt since the coke was detected in one additional image; producing a total of 53 images that represent 89.83% of the total, see Figure 5. This best individual brought into consideration the reflectance, which is a feature that in some images of the can is useful for the solution of the problem. Next, the functions obtained by the OGP are listed.

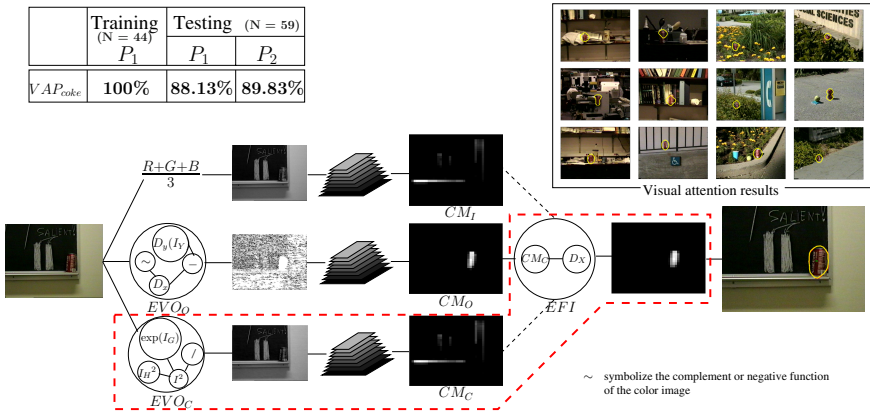
$$\begin{aligned}
 EVO_O &:= Dx(I_K) - Dy(I_Y) \\
 EVO_C &:= (Exp(I_G)/(I_H)^2)^2 \\
 EFI &:= Dx(CM_C)
 \end{aligned}$$

As a final experiment, the training stage was applied to the  $FOA_{signal}$  in order to detect the object of interest, in this case the traffic signal, with a percentage rate of 88.89%; in other words, in 40 of the 45 training images. Next, during the testing stage the  $FOA_{signal}$  detected the object of interest, traffic signal, in 77.78% of the images. Thus, from 45 testing images the best evolved program correctly detected





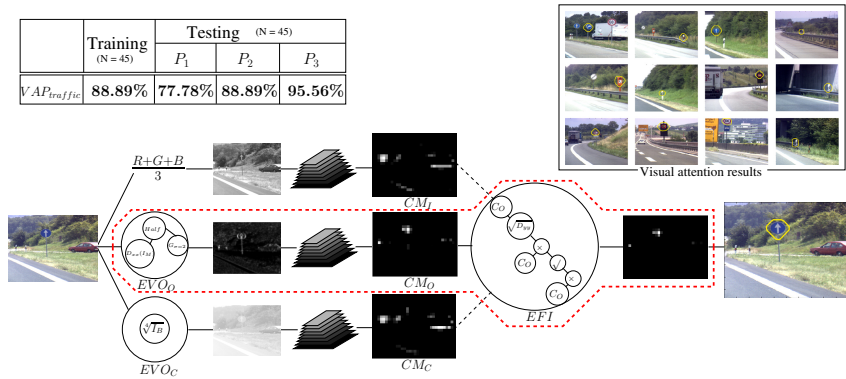
**Fig. 4** These figures show the average, highest, and lowest fitness; for a) the red can run, and b) the traffic signal that produce the respective best individual of Figures 5 and 6



**Fig. 5** Top-Down testing for the red can problem. Evolved structure of  $VAP_{Can}$  obtained through the OGP to attend the red can in the images

the salient object in 35 images. Moreover, the percentage of detection increases when we consider a second attempt as the signal was detected in five additional images producing a total of 40 images that represent a rate of 88.89%. Finally, in a third attempt the percentage increases to 95.56%, see Figure 6. Next, the best functions that were discovered by the OGP are listed.

$$\begin{aligned}
 EVO_D &:= \text{Half}(|G_{\sigma=2}(\text{Half}(\text{Half}(G_{\sigma=2}(\text{Half}(|D_{xx}(I_M)|))))))|) \\
 EVO_C &:= \sqrt[4]{I_B} \\
 EFI &:= \left( \sqrt{\sqrt{D_{yy}(CM_O)} \times CM_O} \right) \times CM_O
 \end{aligned}$$



**Fig. 6** Top-Down testing for the traffic signal problem. Evolved structure of  $VAP_{traffic}$  obtained through the OGP to attend traffic signals in the images

## 6 Conclusions

This work presents a new and useful approach for understanding visual attention. The experiments are motivated by new ideas about purposive evolution and organic genetic programming. The results confirm that it is possible to obtain VAPs that fulfill the task at hand. Moreover, an original program that solves the size pop-out search task was obtained by our approach, and to our knowledge it is the first time to be achieved. Also, the incorporation of shape dimension, carried out with morphological operations, is an original contribution to the research in visual attention.

## References

- Baluch, F., Itti, L.: Mechanisms of top-down attention. *Trends in Neurosciences* 34(4), 210–224 (2011)
- Barton, R.A.: Visual specialization and brain evolution in primates. *Proceedings of the Royal Society of London Series B-Biological Sciences* 265(1409), 1933–1937 (1998)
- Borji, A., Sihite, D., Itti, L.: Computational modeling of top-down visual attention in interactive environments. In: *Proceedings of the British Machine Vision Conference*, pp. 85.1–85.12. BMVA Press (2011)
- Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience* 3(3), 201–215 (2002)
- Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. *Annual Reviews* 18, 193–222 (1995)
- Feldman, J.A.: Four frames suffice: A provisional model of vision and space. *Behavioral and Brain Sciences* 8(02), 265–289 (1985)
- Frintrop, S.: *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*. LNCS (LNAI), vol. 3899. Springer, Heidelberg (2006)
- Gottlieb, J.: From thought to action: the parietal cortex as a bridge between perception, action, and cognition. *Neuron* 53(1), 9–16 (2007)

9. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology* 148, 574–591 (1959)
10. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(11), 1254–1259 (1998)
11. Kant, I.: *Critique of the power of judgment*. Cambridge University Press (2000)
12. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology* 4(4), 219–227 (1985)
13. Lennox, J.G.: Darwin was a teleologist. *Biology and Philosophy* 8, 409–421 (1993)
14. Milanese, R.: Detecting salient regions in an image: from biological evidence to computer implementation. PhD thesis, Department of Computer Science, University of Genova, Switzerland (December 1993)
15. Milner, D., Goodale, M.A.: *The visual brain in action*, 2nd edn. Oxford University Press (December 1995)
16. Oliva, A., Torralba, A., Castelano, M., Henderson, J.: Top-down control of visual attention in object detection. In: *International Conference on Image Processing*, vol. 1, pp. 253–256 (2003)
17. Peters, R.J., Itti, L.: Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, MN (June 2007)
18. Posner, M.I., Snyder, C.R., Davidson, B.J.: Attention and the detection of signals. *Journal of Experimental Psychology* 109(2), 160–174 (1980)
19. Rensink, R.A.: The dynamic representation of scenes. *Visual Cognition* (2000)
20. Rensink, R.A.: Seeing, sensing and scrutinizing. *Vision Research* 40, 1469–1487 (2000)
21. Treisman, A.: *The Cognitive Brain*. MIT Press, Cambridge (1991)
22. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. *Cognitive Psychology* 12(1), 97–136 (1980)
23. Ungerleider, L.G., Mishkin, M.: Two cortical visual systems. In: Ingle, D.J., Goodale, M.A. (eds.) *Analysis of Visual Behavior*, pp. 549–585. MIT Press, Cambridge (1982)
24. Walther, D., Koch, C.: Modeling attention to salient proto-objects. *Neural Networks* 19(9), 1395–1407 (2006)

# Comparison of Two Evolutionary Approaches for the Polygonal Approximation of Digital Curves

Paola Berenice Alvarado-Velazco and Victor Ayala-Ramirez

**Abstract.** In this paper, we address the approximation of digital curves using straight-line segments. Our objective is to compare the performance of two population-based evolutionary algorithms: an evolutionary programming approach and a variable length chromosome genetic algorithm to solve the polygonal approximation problem. We describe the main elements of the methods under comparison and we show the results of the tests executed on a dataset comprising curves that exhibit a range of conditions with respect to two main features: openness and straightness. Our experiments show that the evolutionary programming based technique is faster and more accurate than the genetic algorithm based approach.

## 1 Introduction

Polygonal approximation of digital curves is a problem addressed by a large number of researchers because it has multiple applications. For example, its use for image compression [1], in the simplification of meshes for CAD models [2] and to approximate shapes for context based image retrieval applications [3], among many others. Our interest in the problem is its potential use in commanding the execution of paths by mobile robots and to control the machining of CNC tools directly from image input.

The polygonal approximation problem can be stated in brief as: Given a digital curve composed of  $N$  points, to find a subset formed of  $M$  of them that can be taken as the end points of successive straight line segments in order to minimize the approximation error. Optimal solutions to this problem can be found using dynamic programming approaches. However, it has also been show that these methods exhibit a  $O(n^4)$  complexity order [4]. This complexity order make them of not practical

---

Paola Berenice Alvarado-Velazco · Victor Ayala-Ramirez  
Universidad de Guanajuato DICIS, Electronics Engineering Department.  
Carr. Salamanca-Valle Km. 3.5+1.8, Salamanca, Mexico, 36700  
e-mail: [palvarado@laviria.org](mailto:palvarado@laviria.org), [ayalav@ugto.mx](mailto:ayalav@ugto.mx)

use for problems where the curves being approximated are composed of a large number of points, as it is the case in most real world applications. That is why meta-heuristics are interesting to approximate digital curves with straight-line segments.

The work in this paper shows the implementation of an evolutionary programming approach to address the polygonal approximation of digital contours. We show details of the elements of the method and we compare its performance against an implementation using variable length chromosome genetic algorithms. Results shows that the Evolutionary Programming based technique is faster to find a solution than the Genetic Algorithm counterpart. Both approaches share the same individual encoding and fitness function.

## 2 Polygonal Approximation of Digital Curves

The problem of the polygonal approximation of digital curves can be stated formally as follows. Let us consider a digital curve  $S$  consisting of  $N$  points,  $S = \cup\{p_i\}$ , that does not consider any order relation among the points  $p_i \in S$ , and with each point  $p_i = (x_i, y_i)$ . The formal problem consists in finding an ordered set of points  $T = \{p_{U(0)}, \dots, p_{U(M)}\}$ ,  $U(k) \in \{1, \dots, N\}$ ,  $k \in \{0, \dots, M\}$  and  $T \subset S$ , such that

$$\begin{aligned} & \arg \min_{T \subset S} |T| \\ & \text{subject to } \|S - T\| < \varepsilon \end{aligned} \quad (1)$$

with  $\varepsilon$  being a predefined error tolerance and  $|T| = M + 1$  being the cardinality of the set  $T$ . Points in  $T$  represent the end points of  $M$  straight line segments that are part of the optimal polygonal approximation.

## 3 Polygonal Approximation of Digital Curves Using Evolutionary Meta-heuristics

There are several works that use evolutionary approaches to solve the polygonal approximation of digital curves. Among them, we can find the use of Genetic Algorithms (GA) [5][6][7], Ant Colony Optimization (ACO) algorithms [8], and Particle Swarm Optimization (PSO) algorithms [9][10][11]. We discuss here several issues on the use of Evolutionary Algorithms (EA) to solve the aforementioned problem. We will show, in particular, an implementation of Evolutionary Programming (EP) used to solve the problem. EP is a technique where the evolution of the individuals is provoked entirely by the mutation operator used, and there is no crossover among the individuals [12]. EP is well suited for the polygonal approximation of digital curves because it can handle variable length chromosomes encoding solutions composed of a variable number of straight-line segments.

## 4 EP Approach to Approximate Digital Curves Using Straight Line Segments

We will provide in what follows details of our implementation. In particular, we will explain the main differences of our approach to previous approaches cited above.

### 4.1 Individual Representation

Previous approaches that use evolutionary methods to approximate digital curves encode the individual of the EA as a binary string of length  $N$ . They consider that the curve being approximated is given as an ordered set of successive points, and they exploit this by representing the presence of a particular point in the candidate polygonal approximation by using a 1 in its corresponding position in the binary string of the individual; and using a 0 instead when the point under consideration is not an end point of any of the segments in the encoded solution. Our proposal to encode a candidate solution for the digital curve approximation problem is to use an integer string composed of  $|T|$  position indexes. These indexes are related to the position of each point in the input information of the digital curve. The encoded solution is then the concatenation of the line segments that join the points  $p_{U(k)}$  and  $p_{U(k+1)}$  for  $k \in \{0, \dots, M-1\}$ .

### 4.2 Mutation Operators

Mutation serves in EP to evolve the population by generating new individuals from those belonging to the current population. Each individual in the EP population generates an offspring by using one of the five mutation procedures provided in our implementation. The proposed mutation operators  $M_i$ , with  $i \in \{1, \dots, 5\}$  are described as follows:

- $M_1$  Mutation on  $n$  bits. The new individual conserves the bit length of its parent but  $n$  bits are mutated. The bits to be mutated are chosen randomly.
- $M_2$  Add a point at the end. This mutation operator adds a randomly chosen segment to the end of the bit string of its parent. So, the bit string length is augmented in  $N_B$  bits.
- $M_3$  Remove a point at the end. Using this operator, the offspring is generated by deleting its ending point. The bit string length of the offspring is reduced  $N_B$  bits with respect to that of the parent.
- $M_4$  Insert a point. The mutated individual is a modified copy of the parent where one of the segments of the parent is broken into two separated segments that are joined at the inserted point. Bit string length of the offspring is  $N_B$  longer than that of the parent.

- $M_5$  Remove a point. This mutation operator removes a point in a random position of the segment chain of the parent individual. After that, it links previous point in the chain to the following point into one segment. The bit length of the string that encodes the mutated individual is reduced in  $N_B$  bits.

In order to choose which mutation operator to apply for each individual, we use a roulette-wheel approach. That is, we assign to each mutation operator  $M_i$  an *a priori* probability  $P_i$  representing our expertise for solving the polygonal approximation problem. According to this, we have assigned to  $M_1$  the largest probability of the mutation operator because it is the operator that helps the most to conserve diversity in the population. The probabilities for all the mutation operators are presented in Table 1.

**Table 1** Probabilities used to choose a mutation operator from the available operators

i	$P_i$
1	0.32
2	0.17
3	0.17
4	0.17
5	0.17

### 4.3 Fitness Function

The fitness score results of the aggregation of six measures that try to ensure that the good individuals exhibit two main characteristics:

- (i) The existence of points of the synthesized polygonal curve in the figure being approximated; and,
- (ii) the coverage of the entire digital curve through the synthesized polygonal approximation.

These measures are described below:

- $F_1$  Overall Distance to Target Curve. For each encoded solution, we consider the sum of distances from the line segments composing it, to the actual curve points in the image. Only a sample of points in these segments is used to reduce computational load.
- $F_2$  Non-collinearity Factor. This factor is used to prevent selection of collinear straight line segments as consecutive segments of the polygonal approximation. Its value depends on the angle between consecutive segments.
- $F_3$  Vertex Over Crowding Factor. This factor penalizes individuals that exhibit a large number of vertices in small regions.

- $F_4$  Bounding Box Factor. This factor favors the coverage of the bounding box area of the curve being approximated by the encoded solution.
- $F_5$  Length Similarity Factor. This factor favors the encoded solution that exhibit a length close to the length of the curve being approximated.
- $F_6$  Recall Factor. The recall factor measures the number of points in the reference curve that are present in the encoded solution or in a given tolerance radius.

The overall fitness function is then:

$$F = (1 + k_6 F_6)(k_1 F_1 + k_2 F_2 + k_3 F_3 + k_5 F_5) + k_4 F_4 + k_7 F_6 \quad (2)$$

The constants  $k_i$ ,  $i \in \{1, \dots, 7\}$  in Equation (2) were calculated automatically in a parameter-tuning test by using a Genetic Algorithm. A circle image was chosen as the reference figure because it was a base case along the design process. The values of these parameters are presented in Table 2.

**Table 2** Parameters of the fitness function

i	$k_i$	i	$k_i$
1	2.00	5	4.35
2	47.31	6	36.46
3	20.38	7	3.91
4	0.24		

## 5 Tests and Results

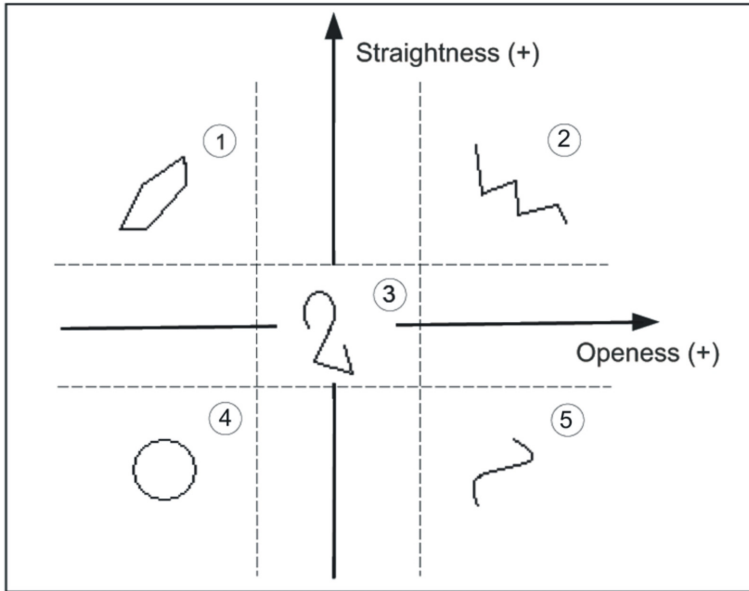
In this section, we first talk about the features of the dataset used to test the proposal. Next, we present some results on this dataset.

### 5.1 Test Protocol

The objective of our tests is to observe the behavior of the EP based approach under different error thresholds and to compare it with that shown by a GA based approach. We have recorded the number of generations needed to achieve a given error threshold and the optimal number of segments found by each evolutionary method.

Our dataset consists of five figures that were chosen to sweep a range of openness and straightness features as shown in Figure 1. Openness refers to how closed or open is a figure, while straightness refers to the smoothness in the transitions between the component segments.





**Fig. 1** Test set of figures exhibit a range of the openness and straightness properties

We have used as performance metric the distance error, which is the average distance from each pixel in the resulting figure to the nearest pixel in the reference one. The approximated minimal distance errors that can be achieved for each figure in the dataset were found in [13]. We use these error bounds as the stopping condition for both the AG and EP algorithms in order to make them comparable among themselves. We have also used larger error bounds to qualitatively see the shape of the optimal polygons found by the evolutionary approaches under comparison. Each test has been run 20 times using a fixed number of 600 individuals as suggested by the work previously cited. Both the GA and EP based approaches share the same fitness function, as described above.

## 5.2 Results

The quantitative results obtained from the execution of the tests are presented in Table 3. We can observe that the EP based approach outperforms the GA based method on all the test curves. The GA needs longer a larger number of generations to achieve the same error bound than the EP. In the test figure labeled 3 (see Figure 1), the GA is not able to converge in a maximal number of 1000 generations.




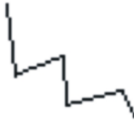
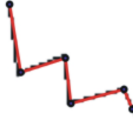
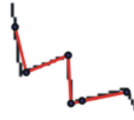







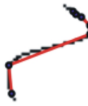

With respect to the number of vertices we can see that, even if both methods increase the optimal number as the error tolerance is reduced, they are kept in a small figure. The catalog of mutation operators can explain this fact, because the individuals evolve by adding vertices to their ends.

**Table 3** Comparison of the quantitative performance of GA and EP approaches for the polygonal approximation of digital curves

Setup		GA		EP	
Figure	Error Bound	$\bar{N}_{gen}$	$\bar{M}$	$\bar{N}_{gen}$	$\bar{M}$
1: $\epsilon^* = 0.66$	$\epsilon < 2\epsilon^*$	54.6	6.35	6.60	7.10
	$\epsilon < 1.5\epsilon^*$	172.5	6.15	11.90	6.80
	$\epsilon < \epsilon^*$	694.75	6.05	22.85	6.90
2: $\epsilon^* = 1.10$	$\epsilon < 2\epsilon^*$	502.75	6.50	17.25	7.50
	$\epsilon < 1.5\epsilon^*$	604.80	6.30	22.15	7.45
	$\epsilon < \epsilon^*$	872.25	6.10	124.95	8.10
3: $\epsilon^* = 0.60$	$\epsilon < 2\epsilon^*$	676.00	5.80	131.75	9.15
	$\epsilon < 1.5\epsilon^*$	1000.00*	-	361.20	10.80
	$\epsilon < \epsilon^*$	1000.00	-	599.45	12.55
4: $\epsilon^* = 1.06$	$\epsilon < 2\epsilon^*$	50.75	7.65	6.05	7.50
	$\epsilon < 1.5\epsilon^*$	267.80	8.15	15.15	8.10
	$\epsilon < \epsilon^*$	938.10	8.50	25.00	9.30
5: $\epsilon^* = 1.64$	$\epsilon < 2\epsilon^*$	1.75	6.25	1.20	5.60
	$\epsilon < 1.5\epsilon^*$	1.80	5.25	1.45	5.65
	$\epsilon < \epsilon^*$	46.10	5.30	3.55	6.10

In the case of the GA, we have used a variable length chromosome that linked parts of well-fitted individuals into a chromosome without constraining it to have the same length of their parents. It is known that the crossover operation is difficult for this variant of GA, so more work is need to find a better crossover operator for the polygonal approximation problem.

In Figure 2, we present the qualitative results for both approaches. We can observe there the finer approximation achieved by the EP approach. We can also see the failure of the GA based approach to converge for the test case number 3.

Figure	Reference	EP	AG
1			
2			
3			
4			
5			

**Fig. 2** Qualitative results of both GA and EP based approaches.

## 6 Conclusions

We have compared an EP based technique and a variable chromosome length GA method with respect to its performance to solve the polygonal approximation of digital curves. The EP based approach exhibits a better performance than the GA based approach with respect to the number of generations needed to achieve a predefined error bound. We have used a test set that covers a range of values for two main features of the digital curves: openness and straightness.

The advantage of the EP comes from the catalog of mutation operators used to perturb good individuals at each epoch. We need to perform more work to improve this catalog.

With respect to the variable chromosome length GA, we need to explore other crossover strategies to improve its overall performance.

The fitness function used by both methods could also benefit from its simplification. We will work in the near future in this direction.

**Acknowledgements.** Paola Berenice Alvarado-Velazco gratefully acknowledges the financial support of her work from Mexico CONACYT through the scholarship 290604/368399.

## References

1. Kolesnikov, A.: Fast algorithm for error-bounded compression of digital curves. In: Proc. of the 2010 IEEE Int. Conf. on Image Processing, Hong Kong, pp. 1453–1456 (2010)
2. Wang, Z., Li, H.Y.: User-controlled geometric feature preserving simplification. In: Fifth International Conference on Frontier of Computer Science and Technology, pp. 335–340 (2010)
3. Sirakov, N., Swift, J., Misna, P.: Image database query using shape-based boundary descriptors. In: Proc. of International Conference on Intelligent Systems Design and Applications (ISDA 2003), pp. 373–382 (2003)
4. Perez, J., Vidal, E.: Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters* 15, 743–750 (1994)
5. Tsai, Y.H.: Fast polygonal approximation based on genetic algorithms. In: Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, pp. 322–326 (2006)
6. Traver, V.J., Recatalá, G., Iñesta, J.M.: Exploring the performance of genetic algorithms as polygonal approximators. In: 15th International Conference on Pattern Recognition, pp. 766–769 (2000)
7. Guanghui, L., Chuanbo, C.: A new approach for polygonal approximation of shape contours using genetic algorithms. In: Second IEEE Conference on Industrial Electronics and Applications, pp. 763–768 (2007)
8. Yin, P.Y.: Ant colony search algorithms for optimal polygonal approximation of plane curves. *Pattern Recognition* 36, 1783–1797 (2003)
9. Dong, F., Xiao, R., Zhong, Y.: An approach to polygonal approximation of digital curves based on discrete particle swarm algorithm. *Journal of Universal Computer Science* 13(10), 1449–1461 (2007)
10. Yin, P.Y.: Genetic particle swarm optimization for polygonal approximation of digital curves. *Image Processing, Analysis, Recognition and Understanding* 16(2), 223–233 (2006)
11. Yin, P.Y.: Vision Systems: Segmentation and Pattern Recognition. Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics, 451–464 (2007)
12. Fogel, L.J.: Intelligence through simulated evolution: forty years of evolutionary programming. John Wiley & Sons (1999)
13. Alvarado-Velazco, P.B., Ayala-Ramirez, V., Sanchez-Yanez, R.E.: Polygonal approximation of digital curves using evolutionary programming. *Acta Universitaria* 22(NE-1), 12–20 (2012)

# Evolution of Contours for Topology Optimization

Gideon Avigad, Erella Matalon Eisenstadt, Shaul Salomon,  
and Frederico Gadelha Guimar

**Abstract.** Topology optimization is used to find a preliminary structural configuration that meets a predefined criterion. It involves optimizing both the external boundary and the distribution of the internal material within a structure. Usually, counters are used a posteriori to the topology optimization to further adapt the shape of the topology according to manufacturing needs. Here we suggest optimizing topologies by evolving counters. We consider both outer and inner counters to allow for holes in the structure. Due to the difficulty of defining a reliable measure for the differences among shapes, little research attention has been focused on simultaneously finding diverse sets of optimal topologies. Here, niching is implemented within a suggested evolutionary algorithm in order to find diverse topologies. The niching is then embedded within the algorithm through the use of our recently introduced partitioning algorithm. For this algorithm to be used, the topologies are represented as functions. Two examples are given to demonstrate the approach. These examples show that the algorithm evolves a set of diverse optimal topologies.

## 1 Background

The background for the methodology of this paper includes genetic algorithms (see Section 1.1), which serve here as the search algorithm, and their use for optimizing topologies (see Section 1.2). Because we aim at finding multiple solutions to a single objective problem, multi-modal optimization through the utilization of niching is of interest. Therefore, a review of niching approaches is given in Section 1.3. Finally, our recently suggested approach to find diverse sets of functions is briefly discussed

---

Gideon Avigad · Erella Matalon Eisenstadt · Shaul Salomon

Department of Mechanical Engineering, Ort Braude College, Karmiel, Israel

e-mail: [gideon, erella, shaulsal}@braude.ac.il](mailto:{gideon, erella, shaulsal}@braude.ac.il)

Frederico Gadelha Guimar

Department of Electrical Engineering, Federal University of Minas Gerais, Brazil

e-mail: [fredericoguimaraes@ufmg.br](mailto:fredericoguimaraes@ufmg.br)

in Section 1.4. In the paper, this approach will serve to enhance the search towards diverse topologies.

## 1.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) belong to a class of non-gradient methods that have grown in popularity following the original publication of [20], and later [8]. expanded the idea and helped make it popular. EAs are stochastic search methods that mimic natural biological evolution. EAs operate on populations of potential solutions, applying the principle of survival of the fittest to produce better and better solutions. An EA uses a population of individuals (solutions) instead of a single solution to perform a parallel search in the problem space. At each generation, a new set of approximations is created by a nature-inspired process. The natural processes commonly mimicked by EAs include selection, breeding, mutation, migration, and survival of the fittest.

## 1.2 EC for Size/Shape/Topology Optimization

According to [15], the problems addressed by structural optimization can be divided into three major categories, namely: a. Topology (layout) Optimization where the search is for an optimal material layout; b. Shape Optimization, where the search targets the optimal contour, or shape, of a structural system whose topology is fixed; and c. Sizing Optimization, in which the optimization aims at optimal cross-sections, or dimensions, of elements of a structural system whose topology and shape are fixed. Finding a good structural configuration (topology) prior to shape and sizing optimization is an important but difficult task. In comparison to shape optimization, topology optimization is more complex since it involves the optimization of both the external boundary and the distribution of the internal material within a structure. Topology optimization is used to find a preliminary structural configuration that meets a predefined criterion. Occasionally it yields a design that can be completely new and innovative. An EC approach to the continuum topology optimization design problem based on genetic algorithms has been proposed, e.g., in [21] and in [22]. In these studies, the design domain was discretized into small elements containing materials or voids in a cantilever plate so that the structure's weight was minimized subject to displacement and/or stress constraints. A variety of structural design fitness functions, among them stiffness, area, and perimeter, were employed to find optimal cantilevered plate topologies. Another coding approach for topologies is based on graph theory [23], in which a topology is represented by a connected simple graph consisting of vertices and simple undirected cubic Bezier curves with varying thicknesses. The derived results show that the graph representation EA can generate clearly defined and distinct geometries and perform a global search. A bit-array representation EA [24] was also implemented

for topology optimization. Design connectivity and constraint handling were further developed to improve the efficiency of the EA. In addition, a violation penalty method has been proposed to drive the EA search towards those topologies with higher structural performance, less unusable material and fewer separate objects in the design domain. A multi-EA system [25] and a variable chromosome length genetic algorithm [16] were proposed for continuum structure topological optimization. Recently, a two-stage adaptive genetic algorithm (TSAGA) [4] was developed in bit-array represented topology optimization. The authors demonstrate the efficiency and effectiveness of TSAGA in comparison to other approaches in reaching global optimal solutions on several case problems.

### ***1.3 Niching within EC and for Topology Optimization***

The sharing method, originally suggested by Holland, [8], is probably the most popular niching technique. Sharing is analogous to a situation in nature in which the resources of a niche have to be shared. In mathematical terms this method penalizes solutions that are similar by dividing the fitness of the niche among them. According to [19], niching methods can be divided into iterative methods, explicit parallel sub-population methods and implicit parallel sub-population methods. Iterative methods address the problem of locating multiple optima of a multi-modal function by repeatedly applying the same optimization algorithm. Several techniques have been used to avoid iterations towards local minima, such as the Tabu technique [7], the sequential niche technique [4] and various jump techniques [12]. Explicit parallel sub-population methods attempt to generate multiple solutions to a multi-modal optimization problem by dividing a population into sub-populations that evolve in parallel. These methods include Multiple-National EA [25], Island EAs [5], the Adaptive Isolation Model [5], and Particle Swarm Optimization [18]. Without communications among the populations, these methods are similar to iterative methods. Implicit parallel sub-population methods attempt to generate multiple solutions by introducing niche/speciation techniques so that population diversity is maintained and many niches survive in a single population. Among these methods are crowding [22], [20], fitness sharing [8], restricted tournament selection [13], species conservation techniques [20] and Genetic Sampler [20]. Species conservation is a relatively new technique for solving multi-modal optimization problems and has been proved effective for obtaining multiple solutions of tested multi-modal problems (e.g., [22]). The study reported in [10] proposed a new approach for finding diverse solutions to a multi-modal problem. The authors suggested posing the single multi-modal problem as a bi-objective problem, with the value of the function as the first objective and the number of neighboring points that are better than others as the second objective. It should be noted that the above studies deal with single objective problems. Although the notion of resource sharing is also an essential part of evolutionary multi-objective optimization this aspect is beyond the scope

of the current paper. Niching within topology optimization has received very little research attention. The few existing studies include the search for optimal topologies for trusses and space planners (see e.g., [2], [17]). There is no record of a simultaneous search for several optimal beam cross sections (topologies) using niching within evolutionary computation.

## 1.4 Function Diversity

The partitioning of the set of functions into subsets (Avigad et al., 2012) is explained in the following (taken from [1]). This partitioning serves as the basis for the approach taken in this paper.

Let  $A = \{f_c(t)\}_{(c \in \Omega, t \in D)}$ , be a set of real valued alternative functions, each sampled  $k$  times, with  $c = [c_1, \dots, c_p]^T$ ,  $\Omega \in \mathbb{R}^p$ ,  $D \in \mathbb{R}$  and  $f_c(t) = [f_c(t_1), \dots, f_c(t_k)]^T$ .  
 $a_j = \min_{c \in \Omega} f_c(t_j)$ ,  $b_j = \max_{c \in \Omega} f_c(t_j)$ ,  $\Delta_j = b_j - a_j$ ,  $j = 1, \dots, k$ ,  $dA = \max_{j=1, \dots, k} \Delta_j$

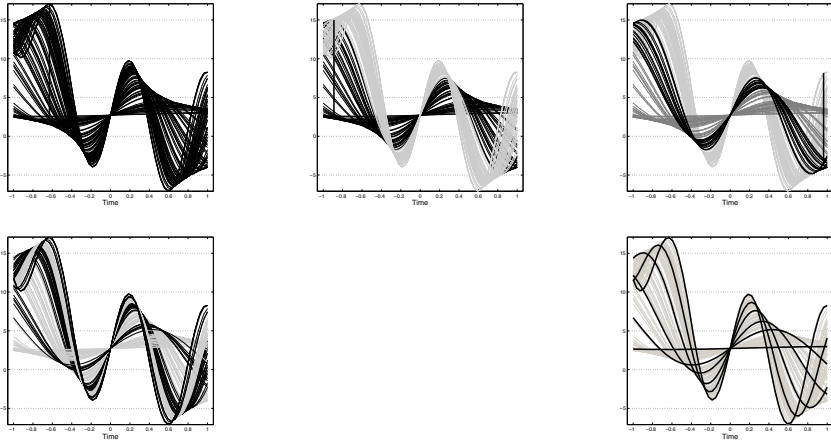
In fact,  $dA$  is the diameter of the set  $A$  in the Chebyshev metric. The idea of the suggested algorithm is that at each step, the set with the largest diameter is partitioned into two subsets. The result of such a partitioning is that subsets are formed, each with a smaller partition than the former maximal partition. This process is repeated until the desired number of subsets is attained. The following algorithm describes the suggested procedure.

### *Partitioning the set $A$ to $m$ subsets*

- (a)  $A_1 \leftarrow A$
- (b)  $i \leftarrow 1$
- (c)  $S \leftarrow \{A_1, \dots, A_i\}$
- (d) **While**  $i < m$  **do**
- (e) Find  $A_j$  such that  $dA_j = \max_{l=1, \dots, k} dA_l$
- (f) Find sample  $t_p$  such that  $\Delta_p = dA_j$ . assemble subsets  $A_{j1}, A_{j2}$  from functions  $f_c(t) = [f_c(t_1), \dots, f_c(t_k)]^T$  which satisfy the inequalities  $f(t_p) \leq a_p + dA_j/2$  and  $f(t_p) > a_p + dA_j/2$  respectively
- (g)  $S \leftarrow \{A_1, \dots, A_{j1}, A_{j2}, \dots, A_i\}$
- (h)  $i \leftarrow i + 1$
- (i) **End while**

To demonstrate the approach, [1] used an artificial set of 100 functions and divided it into six different sub-sets by using the partitioning algorithm. Figure 1 (borrowed from [1]) shows the results of the first, second, third and fifth partitioning stages (resulting in six subdivisions of the function space). As a last step one function is chosen from each sub-division. In [1], such functions were evolved by using set-based





**Fig. 1** Partitioning algorithm

dominance and by implementing crowding, which was based on the number of functions present in each sub-division. The current study proposes evolving contours to optimize topologies by utilizing both the partitioning algorithm and crowding, as suggested in [11].

Evolving contours for the purpose of evolving topologies differs from the claim (see Figure 1) that contours are used only for shape optimization. Moreover, the purpose here is not merely to find one optimal topology but rather to look for several diverse optimal topologies. The motivation for finding different solutions to an optimization problem is well known (see e.g., Mattson and Messac, 2005). Basically, finding different solutions provides decision makers with greater flexibility in choosing the preferred design based on unmodeled properties.

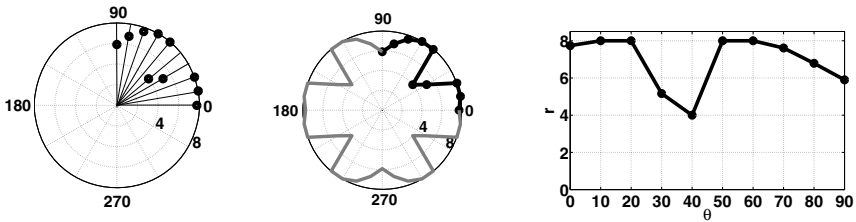
## 2 Methodology

To enhance topology diversity, the algorithm proposed in [11] is adopted and adapted in this paper. The algorithm in [11], deals with functions. This means that in order to use that algorithm for topologies, the topologies should also be described by functions.

### 2.1 Describing Topologies by Functions

In this paper, 2D topologies are considered. Polar coordinates are used here for describing topologies, although, as discussed, other coordinate systems may be used. Topologies with and without holes are considered.

For topologies without a hole, the  $j$ -th contour is described by an ordered set of radii:  $r_j = [r_1^j(\theta_1), \dots, r_{N_\theta}^j(\theta_{N_\theta})]^T$  where  $\theta \in D \subset R$  is an equally spaced angle. The topology is formed by connecting the ordered set to form a continuous function (through linear sections). The left panel of Figure 2 depicts the encoding of a topology with  $k$  points, and the middle panel of that figure depicts the formed topology. It should be noted that in this paper we use reflection symmetry (or mirror symmetry) with respect to the horizontal and vertical axes, although the use of other symmetries or of no symmetry at all may be considered. Reflection symmetry implies that a quarter of the contour may describe the entire contour.



**Fig. 2** The encoding of topology without a hole

In fact, based on these equally spaced coordinates we may now describe the topology as a function of  $\theta$ , so that the  $j$ -th topology may be described as a function  $f_r(\theta) = r(\theta)$ . The middle panel of Figure 2 depicts the topology while the right panel depicts its plot as a function.

Two contours are used to describe a topology with a hole. The  $j$ -th topology is coded by its inner contour radius  $R_j^{in} = [r_1^{j,in}(\theta_1), \dots, r_{N_\theta}^{j,in}(\theta_{N_\theta})]^T$  and by its outer contour radius  $R_j^{out} = [r_1^{j,out}(\theta_1), \dots, r_{N_\theta}^{j,out}(\theta_{N_\theta})]^T$ . The topology is formed by connecting the ordered set to a continuous function (linear sections). The left panel of Figure 3 depicts the code of the topology, where squares and circles designate the inner and outer radii of the inner and outer contours respectively. The middle panel of the figure shows the topology formed by connecting the radii by linear sections and using the rotational reflection.

As in the no-hole case, here also this topology may be described by a function so that the  $j$ -th topology may be described as functions;  $f_{rin}^{rout}(\theta) = \{f^{rout}(\theta) \cup f_{rin}(\theta)\}$ . It should be noted that the outer and inner boundaries of the outer and inner contours should be determined to ensure a reasonable thickness for the topology. The middle panel of Figure 3 depicts the topology, and the right panel shows its description as a function.

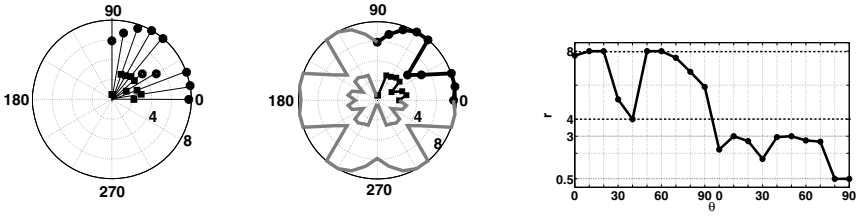


Fig. 3 The encoding of topology with hole

### 2.2 Problem Definition

The optimization problem is defined as follows: For topologies without a hole:

$$\begin{aligned}
 & \min_r F(f_r(\theta)) \\
 & \forall r \in R : r_{min} \leq r \leq r_{max} \\
 & g(f_r(\theta)) \geq 0
 \end{aligned} \tag{1}$$

For topologies with a hole:

$$\begin{aligned}
 & \min_{r^{out}, r^{in}} F(f_{r_{in}}^{r_{out}}(\theta)) \\
 & \forall r^{in} \in R^{in} : r_{min}^{in} \leq r^{in} \leq r_{max}^{in} \\
 & \forall r^{out} \in R^{out} : r_{max}^{in} + \delta \leq r^{out} \leq r_{max}^{out} \\
 & g(f_{r_{in}}^{r_{out}}(\theta)) \geq 0
 \end{aligned} \tag{2}$$

where  $F : F(f(\cdot)) \rightarrow 0 \in \Gamma \subset R$  and  $\Gamma$  is the objective space. For example  $F(f(\cdot))$  can be a cross section area, the first or the second moment of inertia and other topology dependent characteristics.

### 2.3 Diverse Topologies and Niching

Niching is implemented to find a diverse set of topologies. For this purpose, a partitioning algorithm is used. First the following parameters must be computed:  $a_j = \min f(\cdot)$ ,  $b_j = \max f(\cdot)$  and  $\Delta_j = b_j - a_j, j = 1, \dots, k$  as well as  $dA = \max_{j=1, \dots, k} \Delta_j$  then the algorithm of [11] is used by slight changes as follows:

*Partitioning the set A to m subsets*

- (a)  $A_1 \leftarrow A$
- (b)  $i \leftarrow 1$

- (c)  $S \leftarrow \{A_1, \dots, A_i\}$
- (d) **While**  $i < m$  **do**
- (e) Find  $A_j$  such that  $dA_j = \max_{l=1, \dots, k} dA_l$
- (f) Find sample  $\theta$  such that  $\Delta_p = dA_j$ . Assemble subset  $A_{j1}, A_{j2}$  from functions  $f_r(\theta)$  or  $f_{r_{in}}^{out}(\theta)$  which satisfy the inequalities:  
 $f_r(\theta) \leq a_p + dA_j/2$  or  $f_{r_{in}}^{out}(\theta) \leq a_p + dA_j/2$  and  
 $f_r(\theta) > a_p + dA_j/2$  or  $f_{r_{in}}^{out}(\theta) > a_p + dA_j/2$  respectively
- (g)  $S \leftarrow \{A_1, \dots, A_{j1}, A_{j2}, \dots, A_i\}$
- (h)  $i \leftarrow i + 1$
- (i) **End while**

The algorithm partitions a population of topologies (represented as functions) to N predefined sub-sets. A solution will reproduce according to its fitness (value of  $F(f(\cdot))$ ) which will be penalized according to the number of functions with which it shares the same sub-partitioning (same niche). Such a penalty will prevent genetic drift and will increase the chances that other optimal solutions will reproduce. For example, Figure 4 depicts 20 topologies represented by their related

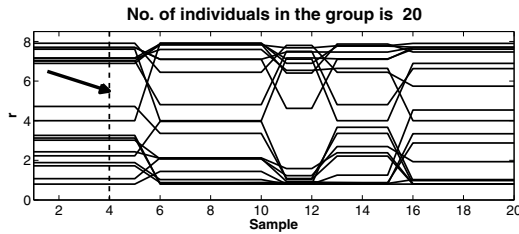


Fig. 4 20 individuals before division

functions. The dashed line marks the sample with the largest difference and therefore the point where the algorithm will divide the group into two groups. This partitioning results in 5 and 15 individuals in each subdivision, as depicted in Figure 5. The next partition is marked by the dashed line in the right panel of Figure 5. The result of dividing the second group (of 15 individuals) into two

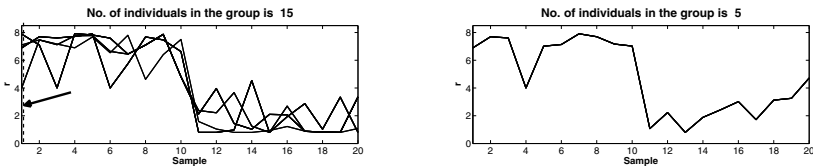


Fig. 5 The two formed subdivisions containing 5 (left panel) and 15 (right panel) individuals

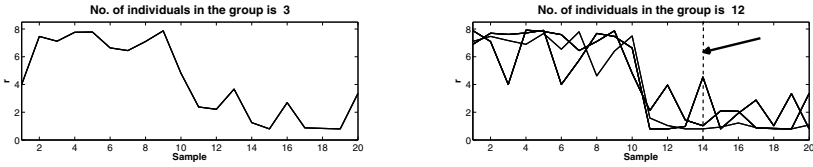


Fig. 6 The second and third groups

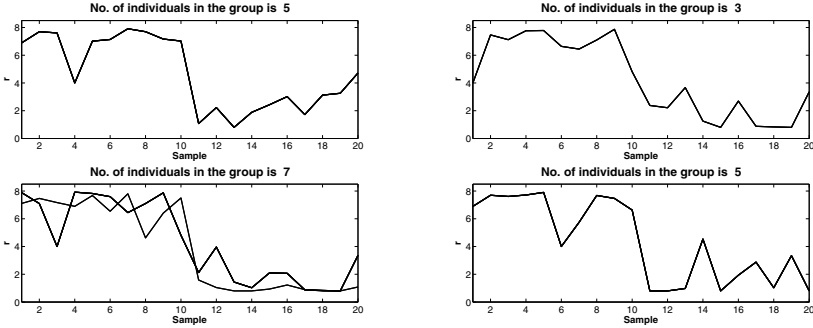


Fig. 7 The second and third groups

groups is depicted in Figure 6. The point where the algorithm will divide again to form the next two groups is marked by a dashed line. Finally, the four groups are presented in Figure 7.

### 2.4 The Evolutionary Optimization Algorithm

The algorithm described here aims at searching for optimal topologies based on the optimization problem set in Section 2.2. The algorithm utilizes an archive in order to preserve elite solutions.

#### The Evolutionary Optimization Pseudo Code

- (a) Initialize a population  $P_t$  with  $n$  individuals. Also, create a population  $Q_t = P_t$
- (b) **While**  $t \leq$  predefined number of generations
- (c) Combine parent and offspring populations and create  $R_t = P_t \cup Q_t$
- (d) For each individual  $z \in R_t$ , compute:
  - (d.1)  $F(f(z))$
  - (d.2)  $g(f(z))$
  - (d.3) The niche count:  $m(z)$
  - (d.4) The fitness:  $fit(z) = F(f(z)) \frac{2n}{2n - m(z)}$

- (e) Create elite population  $P_{t+1}$  of size  $n$  from  $R_t$  (procedure I)
- (f) Create offspring population  $Q_{t+1}^*$  of size  $n$  from  $R_t$  by Tournament selection (procedure II)
- (g) Perform Crossover to obtain  $Q_{t+1}^{**}$  from  $Q_{t+1}^*$ .
- (h) Perform Mutation to obtain  $Q_{t+1}$  from  $Q_{t+1}^{**}$ .
- (i) Set  $t = t + 1$  and go to (b).
- (j) Provide the decision maker with  $N$  topologies, each from a different subdivision and each possessing  $\min(\text{fit})$  within this subdivision.

*Procedure I: Archiving*

Compute  $n_c = \text{number of individuals that comply with } g(f(z)) > 0$

- (a) If  $n_c \leq n$  then include the  $n_c$  individuals in  $P_{t+1}$  and sort the remaining  $2n - n_c$  individuals to a list  $L1(g(f(z) : <))$ . Add the first  $n - n_c$  individuals of  $L1$  to  $P_{t+1}$ .
- (b) If  $n_c > n$ 
  - (b.1) Include  $N$  individuals in  $P_{t+1}$  taking one individual from each sub division that is with the  $\min(\text{over all fit})$ .
  - (b.2) Then list all  $n_c - N$  individuals to a list  $L2(\text{fit}(f(z) : >))$ . Add the first  $n - N$  individuals of  $L2$  to  $P_{t+1}$ .

*Procedure II: Tournament Selection*

- (a) If  $g(f(z)) \geq 0 \wedge g(f(z')) < 0$  include  $z$  in  $Q_{t+1}$
- (b) If  $g(f(z)) < 0 \wedge g(f(z')) < 0$   
Then if  $g(f(z)) > g(f(z'))$  include  $z$  in  $Q_{t+1}$
- (c) If  $g(f(z)) \geq 0 \wedge g(f(z')) \geq 0$   
Then if  $\text{fit}(f(z)) \text{ leqfit}(f(z'))$  include  $z$  in  $Q_{t+1}$

**Algorithm explanation**

In step (a), two populations are initialized. A "while loop" begins at step (b) and ends at step (i). In this loop populations of candidate solutions are evolved. Step (c) is a common step used in MOEAs, which involves creating a combined population. In step (d), the following calculations take place with respect to each individual within a population: (d.1)- value of the objective function (which is to be optimized); (d.2)- value of the constraint; (d.3)- number of functions that share the same subdivision with the individual; (d.4)- fitness value of the individual, which is computed by penalizing the objective function's value. A greater number of shared individuals results in a higher penalty and therefore decreased fitness. In step (e) the elite population is formed by utilizing Procedure I. The procedure admits feasible optimal (low fitness value; without loss of generality) solutions to the archive. Moreover, it ensures that the archive will always include at least one representative from each of the  $N$  subdivisions. In step (f) the offspring population is created by using Procedure II. In each tournament (among two arbitrary selected individuals), the winning individual is feasible and possesses a lower fitness value (without loss of generality)

than its competitor. If both competitors are infeasible, the one with a lesser violation of the constraint is the winner. Steps (g) and (h) are the crossover and mutation steps respectively. In the final step (step (j)), the decision maker is presented with  $N$  solutions that have the lowest fitness value (the best as far as minimization is considered). In order to preserve diversity, each of the presented topologies is extracted from another sub-division. Please note that step (j) is an a posteriori step and the decision maker is not involved within the evolutionary search.

### 3 Examples

In this section two examples are used to demonstrate the proposed approach and to highlight its potential for solving real life topology optimization problems.

#### 3.1 Cross Section Optimization for a Structure Subjected to a Tensile Force

A beam is subjected to a tensile force of magnitude:  $F_r = 150kN$ . The objective of the optimization is to find a topology with a minimal cross section  $A$ . The allowed normal stress is:  $[\sigma] = 10MPa$ , therefore

$$A_{min} \geq \frac{F_r}{[\sigma]} = \frac{150 \times 10^3}{10 \times 10^6} = 150cm^2.$$

This implies on the constraint:

$$g = A - A_{min} = A - 150 \geq 0$$

The search here will be conducted towards topologies that contain a hole. The topology search space is a priori set such that:

$$0.5 \leq r^{in} \leq 3; 3 + 1 = 4 \leq r^{out} \leq 8, \text{ where dimensions are given in cm.}$$

The initial population includes 100 individuals and is represented as functions in the left panel of Figure 8. In order to prevent the outer contour from dominating the diversity, normalization is applied such that the amplitude of both contours is the same. The normalized functions are depicted in the right panel of Figure 8. Note that the radii have not been altered and that the inner radius seems bigger due to the normalization.

Two topologies arbitrarily chosen from the initial population are depicted in Figure 9. These are clearly non-optimal solutions because the topology in the left panel involves a cross section that is much too big according to the constraint, whereas the topology in the right panel has a cross section that is smaller than allowed.

Apart from diversity preservation, the algorithm has evolved optimal topologies. Figures 10 and 11 depict the resulting optimal diversified topologies when the proposed algorithm is run with  $N=2$  and  $N=4$ , respectively. Clearly the algorithm has

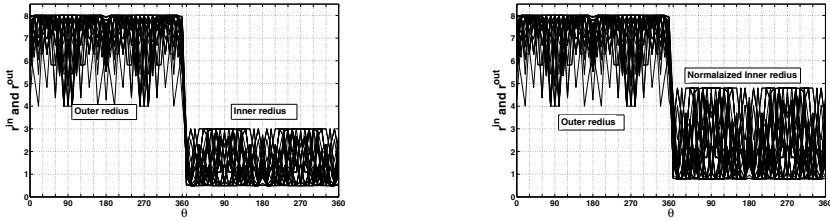


Fig. 8 Normalized and Un-normalized function

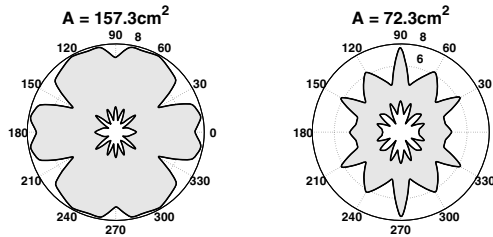


Fig. 9 Two cross section in the first generation

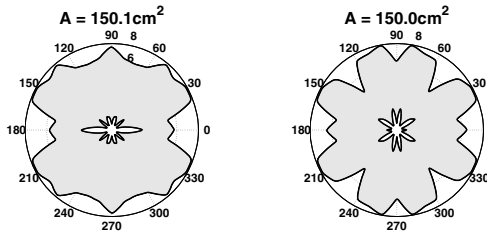


Fig. 10 The result for  $N=2$

evolved diverse optimal topologies. All are distinctly different and possess the minimal (or close to the minimal) possible cross section area. Considering feasibility, clearly most topologies are not natural candidates for manufacturing. Nevertheless, as expected from such algorithms, they have shown the way, and adaptations to facilitate manufacturing should follow. One approach that should help is to encode the topologies with less complexity, i.e., fewer angle divisions. Figure 3.1 depicts the same problem; however, instead of using 10 angle divisions to decode the topologies, 5 divisions are used.



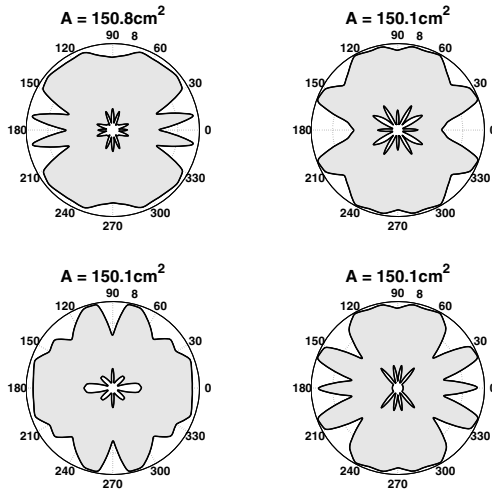
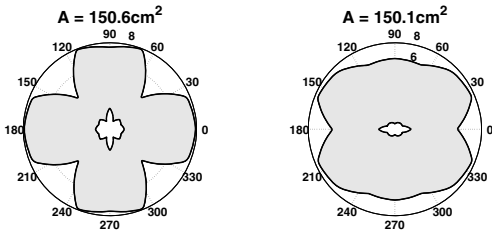


Fig. 11 The result for N=4

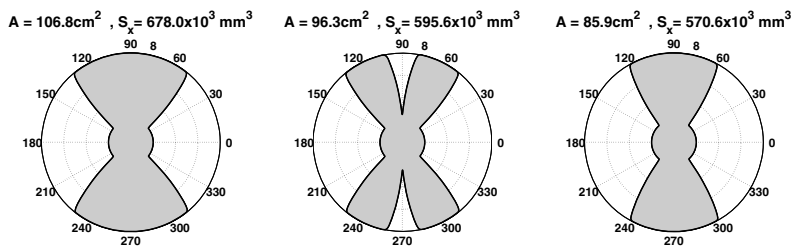


### 3.2 Cross Section Optimization for a Structure Subjected to a Moment

A beam is subjected to a pure moment. The moment applied magnitude is:  $M = 66 \times 10^3 N \times m$ . The allowed normal stress is:  $[\sigma] = 120 MPa$  The equation connecting the allowable stress and moment is:  $[\sigma] \geq \frac{M y_{max}}{I_{xx}} = \frac{M}{S_x}$  where  $I_{xx}$  is the second moment of inertia,  $y_{max}$  is the the maximal perpendicular distance from axis x, and  $S_x = \frac{I_{xx}}{y_{max}}$  is the section modulus about x axis. The allowable stress determined the allowable range of section modulus about the x axis:

$$[\sigma] \geq \frac{M}{S_x} \rightarrow S_x \geq \frac{M}{[\sigma]} = \frac{66 \times 10^3}{120 \times 10^6} = 5.5 \times 10^{-4} m^3 = 550 cm^3$$

The constraint is  $g = S_x - 550 \geq 0$  and the objective is the minimal beam weight. Because the problem deals only with constant cross section beams, minimizing the weight is the same as minimizing the beam’s cross section area. Figure 12 depicts three evolved topologies with no holes. It is apparent that although different



**Fig. 12** The result for  $N=3$  decoded without a hole

topologies have been evolved, their cross sections are not similar. As expected, the optimal topology involves an I-shaped cross section, where other topologies are less optimal versions of it.

## 4 Summary and Conclusions

In this paper, we have adapted a recently proposed partitioning algorithm in order to evolve a diverse set of optimal topologies. It has been suggested to code topologies using polar coordinates. Topologies with holes and without holes were coded. The coding was used to establish topologies through linear interpolation. Then, the topologies were represented as functions (of an equally spaced angles). Once this was achieved, the partitioning algorithm was implemented within an evolutionary search. This algorithm enhances a search toward a diverse set of optimal topologies. These diversified optimal topologies are associated with multiple optimal solutions (if the problem is inherently multi modal) or with different levels of optimality (if the problem is not multi modal by nature). In future work other coordinates will be used to describe the topologies (Cartesian), allowing more complicated structure with more holes. Moreover, 3D shapes should be evolved by describing the added dimension using a function as well. Real life problems should be considered, and comparisons with other approaches should be made.

**Acknowledgements.** This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.

## References

1. Avigad, G., Goldvard, A., Salomon, S.: Partitioning Algorithms. Technical report No. br352012, <http://brd.braude.ac.il/~gideon>
2. Conceio, A.: A hierarchical genetic algorithm with age structure for multimodal optimal design of hybrid composites. *Structural Multidisciplinary Optimization* 31, 280–294 (2006)

3. Erlbaum, L.: Conference on Genetic Algorithm, pp. 41–49. Associates Inc., Hillsdale (1987)
4. Glover, F.: Tabu search part I. *ORSA Journal on Computing* 1(3), 190–206 (1989)
5. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison Wesley, Massachusetts (1989)
6. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the Second International*
7. Hadar, J., Rusell, W.R.: Rules for Ordering Uncertain Prospects. *American Economic Review* 59, 25–34 (1969)
8. Holland, J.H.: *Adaptation in neural and artificial systems*. The University of Michigan Press, Ann Arbor (1975)
9. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Eschelman, L. (ed.) *Sixth International Conference on Genetic Algorithms*, pp. 24–31. Morgan Kaufmann, San Francisco (1995)
10. Jahn, J.: *Vector Optimization, Theory, Applications and Extensions*, 2nd edn. Springer, Heidelberg (2011)
11. Jensen, E.D.: *Topological structural design using genetic algorithms*, Phd. Dissertation, Purdue University, Lafayette (1992)
12. Li, J.-P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation* 10(3), 207–234 (2002)
13. Li, J.-P., Li, X.-D., Wood, A.: Species Based Evolutionary Algorithms for Multimodal Optimization. In: *A Brief Review, WCCI 2010 IEEE* (2010)
14. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: *IEEE Int.Conf. on Neural Networks*, pp. 1942–1948 (1995)
15. Kicinger, R., Arciszewski, T., De Jong, K.A.: Evolutionary computation and structural design: a survey of the state of the art. *Computers and Structures* 83(23-24), 1943–1978 (2005)
16. Kim, I.Y., de Weck, O.L.: Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Structural and Multidisciplinary Optimization* 29(6), 445–456 (2005)
17. Kirk Martini, P.E.: Harmony Search Method for Multimodal Size, Shape, and Topology Optimization of Structural Frameworks. *Journal of Structural Engineering* 137(11), 1332–1339 (2011)
18. Mahfoud, S.W.: Crowding and preselection revisited. In: Bnner, R.M., Manderick, B. (eds.) *Proceedings of the Second International Conference on Parallel Problem Solving from Nature - PPSN 2011*, vol. 36, pp. 27–36. Elsevier Science Publishers (1992)
19. Mengshoel, O.J., Goldberg, D.E.: The Crowding Approach to Niching in Genetic Algorithm. *Evolutionary Computation* 16(3), 315–354 (2008)
20. Rechenberg, I.: *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*, Stuttgart. Fommann-Holzbook (1973)
21. Sandgren, E., Jensen, E.D., Welton, J.: Topological design of structural components using genetic optimization methods. In: *Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers*, pp. 31–43 (1990)
22. Stoean, C., Preuss, M., Stoean, R., Dumitrescu, D.: Disburdening the Species Conservation Evolutionary Algorithm of Arguing with Radii. In: *GECCO 2007*, pp. 1420–1427 (2007)
23. Wang, S.Y., Tai, K.: Graph representation for structural topology optimization using genetic algorithms. *Computers and Structures* 82(2021), 1609–1622 (2004)

24. Wang, S.Y., Tai, K.: Structural topology design of optimization using genetic algorithm with a bit-array representation. *Computer Methods in Applied Mechanics and Engineering* 194(3638), 3749–3770 (2005)
25. Woon, S.Y., Tong, L., Osvaldo, O.M., Steven, G.P.: Effective optimization of continuum topologies through a multi-GA system. *Computer Methods in Applied Mechanics and Engineering* 194(3033), 3416–3437 (2005)
26. Li, X.: Adaptively Choosing Neighbourhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization. In: Deb, K., Tari, Z. (eds.) *GECCO 2004*. LNCS, vol. 3102, pp. 105–116. Springer, Heidelberg (2004); *Fommann-Holzbook* 1973

**Part VIII**  
**Real-world Application of Bio-inspired**  
**Metaheuristics**

# Multi-Objective Particle Swarm Optimisation for Molecular Transition State Search

Jan Hettenhausen, Andrew Lewis, Stephen Chen, Marcus Randall,  
and René Fournier

**Abstract.** This paper describes a novel problem formulation and specialised Multi-Objective Particle Swarm Optimisation (MOPSO) algorithm to discover the reaction pathway and Transition State (TS) of small molecules. Transition states play an important role in computational chemistry and their discovery represents one of the big challenges in computational chemistry. This paper presents a novel problem formulation that defines the TS search as a multi-objective optimisation (MOO) problem. A proof of concept of a modified multi-objective particle swarm optimisation algorithm is presented to find solutions to this problem. While still at a prototype stage, the algorithm was able to find solutions in proximity to the actual TS in many cases. The algorithm is demonstrated on a range of molecules with qualitatively different reaction pathways. Based on this evaluation, possible future developments will be discussed.

## 1 Introduction

Many chemical compounds can exist in a number of different forms, or isomers, each having the same molecular formula but different physical and chemical properties,

---

Jan Hettenhausen · Andrew Lewis

Institute for Integrated and Intelligent Systems, Griffith University, Australia

e-mail: [j.hettenhausen,a.lewis}@griffith.edu.au](mailto:{j.hettenhausen,a.lewis}@griffith.edu.au)

Stephen Chen

School of Information Technology, York University, Canada

e-mail: [sychen@yorku.ca](mailto:sychen@yorku.ca)

Marcus Randall

Faculty of Business, Technology and Sustainable Development, Bond University, Australia

e-mail: [mrandall@bond.edu.au](mailto:mrandall@bond.edu.au)

René Fournier

Department of Chemistry, York University, Canada

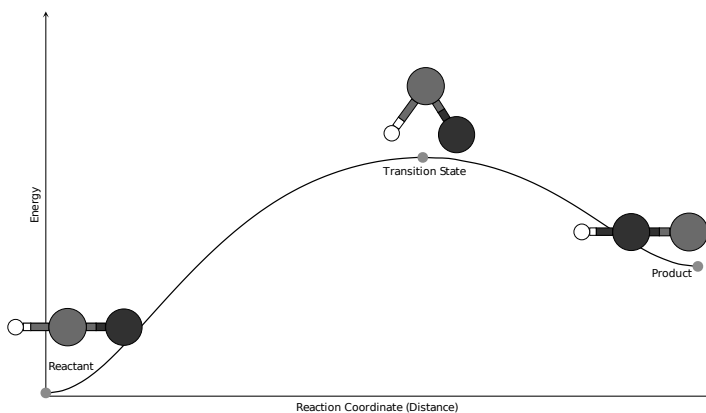
e-mail: [renef@yorku.ca](mailto:renef@yorku.ca)

due to their different molecular structure. The process of isomerisation, changing a molecule from one structure to another, involves passing through an intermediate or *transition state*. This transition state is the unstable intermediate state at which the molecule has the highest potential energy between the two, stable, low-energy isomers. Figure 1 illustrates a reaction path from reactant via transition state to product on the example of hydrogen cyanide.

TS play an important role in a variety of areas of theoretical chemistry. Accurate knowledge of geometry and potential energy of TS is, for example, required to understand chemical reactivity, reaction rates and mechanisms as well as analysing barrier lowering in catalysed reactions [17, 20]. For example, the Arrhenius law,  $k = Ae^{-E_a/RT}$ , gives a good fit to the observed rate constant  $k(T)$  of chemical reactions. The parameters  $A$  and  $E_a$  are readily obtained if one knows the transition state:  $E_a$  is the energy difference between the saddle point and the local minimum, and  $A$  can be estimated from the curvature (energy second derivatives) at the saddle point. As transition states cannot be directly observed and even their approximation by experimental means is difficult, determining transition states computationally has been a field of ongoing research in computational chemistry.

Algorithmically this problem is usually approached by approximating the reaction pathway on the Potential Energy Surface (PES) of the molecule [35]. While generally not known in its entirety, individual points on the PES can be acquired by solving the Schrödinger equation for a specific conformation of the nuclei of the molecule in question. The PES can be pictured as a hilly landscape, in which the stable isomers are valley bottoms (local minima). The transition states then are the passes, i.e. the lowest crossing point over the mountains between two valleys [35].

While for macro-molecules, such as proteins, the search for the various local minima of a molecule is still a challenging problem, it is considerably more difficult to obtain the saddle points even for small molecules. This is in particular due to the fact



**Fig. 1** Approximate illustration of a reaction path showing corresponding conformations of hydrogen cyanide

that TSs are first order saddle-points of the PES and therefore require significantly more complicated approaches than are necessary to identify local minima.

The process of identifying the TS for a given local minimum of a molecule (reactant) comprises five distinct steps (see, for example, Fournier *et al.* [17]):

- (i) Locate the reactant on the PES. Depending on the problem definition it may also be necessary to identify the second local minimum (product) which the reaction will result in.
- (ii) Determine a good approximation for the TS. This is often based on finding an approximation to the reaction pathways and identifying a candidate TS on it.
- (iii) Refine the TS by performing a constraints minimisation of the norm of gradient. Appropriate constraints need to be chosen to ensure that the search does not converge to either of the local minima.
- (iv) Calculate the Hessian matrix and frequencies of the candidate TS to verify that the point is, in fact, a first order saddle-point.
- (v) Map out the intrinsic reaction coordinate.

Step 2 usually represents the most challenging aspect of identifying a TS. Once an approximate TS is identified, it is usually straightforward to find the actual TS. The algorithm presented in this paper addresses the problem of finding such an approximate TS with sufficient accuracy.

A large percentage of TS finding algorithms can be classified as either Shallowest Ascent (SA) or Path Interpolation (PI) methods. SA methods start at the reactant and iteratively try to follow the shallowest ascent out of the valley until the norm of gradient becomes zero and the TS is found. One of the first practical approaches to finding the TS was developed by Cerjan and Miller [5] in 1981. Other approaches were published by, for example, Baker [1] and Maeda and Kono [31]. To identify the shallowest ascent the second derivative matrix of the PES needs to be calculated in every step making SA approaches computationally more expensive than PI methods. Their advantage, however, is that only the first minimum needs to be known. SA approaches therefore solve a more general and more difficult problem than PI approaches, which require *a priori* knowledge of both minima, i.e. the reactant and the product. PI approaches start with a coarse approximate path between the two minima and gradually refines it using constraint optimisation. Common PI methods include nudged elastic band [15, 22, 23, 41], the string method (SM) [13], Growing String Method (GSM) [3, 4, 34, 38] and modified GSM [19, 20]. Further methods outside these two categories include the Fast Marching Method (FFM) [2, 11, 12] and Dewar's method [10]. Comprehensive overviews of existing methods can be found in del Campo and Köster [9] and Schlegel [35].

To the authors' knowledge, the first non-deterministic method was developed by Fournier *et al.* [17]. Their approach combines the ideas of SA with those of common metaheuristics by using a small group of "climbers" that traverse an approximate reaction pathway towards the TS. Climbing direction and movements are thereby guided by a set of parameters acquired from the global information of the path gathered by all climbers. The performance of each climber is assessed with an aggregated fitness function based on energy, distance and changes in bonds. An



interesting feature of this approach is that the climbers can continue after finding an approximate TS, allowing the algorithm to discover successive minima and TS without having to restart the algorithm from a new position.

Based on the idea that the reaction pathway follows the shallowest path out of the valley around the minimum, this paper presents a novel problem formulation for TS search treating the problem as a multi-objective optimisation problem of the following form:

$$\begin{aligned} \text{minimise } \mathbf{f}(\mathbf{x}) &= \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ f_k : \mathbb{R}^n &\rightarrow \mathbb{R}, \forall \mathbf{x} \in \mathfrak{F}, \mathfrak{F} \subseteq \mathfrak{S} \subset \mathbb{R}^n \end{aligned} \quad (1)$$

where  $\mathfrak{F}$  is the feasible space within the search space  $\mathfrak{S}$  based on a set of given constraints. A detailed discussion on this formulation and its justification will be provided in Section 2. Using this problem formulation then allows finding the reaction pathway and TS using multi-objective metaheuristics, a class of algorithms highly successful on a wide range of problems in science and engineering (e.g. [21, 24, 25, 37]). While non-deterministic, metaheuristics have the advantage of being inherently parallel and do not require structural knowledge of the problem domain.

As a solution technique, a prototype of a modified multi-objective particle swarm optimisation algorithm will be used. Particle Swarm Optimisation (PSO) takes its inspiration from the behaviour of swarming and flocking animals [29]. PSO models candidate solutions as a populations of particles that can either freely or with constraints move in the parameter space of the problem. The velocity of the particles is thereby governed by a velocity equation usually based on momentum, a particle specific memory of the best solution visited by that particle and a social component giving each particle knowledge of the best solution known to the swarm. Sometimes a turbulence component is used to improve diversity [39]. A variety of PSO adaptations exist to allow simultaneous optimisation of multiple objectives [6, 16, 32]. The primary difference to single-objective PSO is that multi-objective PSO (MOPSO) incorporates an archive that holds knowledge of a set of known optimal compromise solutions. Optimality in this context is defined on the basis of the *dominance* relation where one parameter vector  $\mathbf{x}_1$  is said to dominate another parameter vector  $\mathbf{x}_2$  if and only if

$$\begin{aligned} f_k(\mathbf{x}_1) &\leq f_k(\mathbf{x}_2), \forall k \in 1, \dots, n_k \\ \exists k \in 1, \dots, n_k &: f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2) \end{aligned} \quad (2)$$

The set of all globally non-dominated parameter vectors form the Pareto-optimal set or

$$\mathfrak{P}^* = \{\mathbf{x}^* \in \mathfrak{F} \mid \nexists \mathbf{x} \in \mathfrak{F} : \mathbf{x} \prec \mathbf{x}^*\} \quad (3)$$

and the set of corresponding objective vectors is called Pareto-front or

$$\mathfrak{P}\mathfrak{F}^* = \{\mathbf{f}(\mathbf{x}^*) \mid \mathbf{x}^* \in \mathfrak{P}\}. \quad (4)$$

which could be described as the set of optimal compromise solutions. For each solution no objective can be further improved without worsening another.

On the basis of the MOPSO algorithm a method will be presented that can successfully converge to the minimum energy pathway. However, a number of adaptations and additions were necessary to address the specific challenges of identifying the reaction on the PES in the context of a multi-objective optimisation problem. These will be discussed in Sections 2 and 3.

The algorithm was tested on a set of small molecules with different reaction pathways. The results of these tests will be discussed in Section 4. Concluding remarks and a discussion of future work and extensions to the method presented can be found in Section 5.

## 2 The Multi-Objective Optimisation Problem Formulation

This paper introduces a novel way of describing the TS search problem by defining the minimum energy pathway section-wise as Pareto-fronts, transforming it into a multi-objective optimisation problem. Based on this formulation, algorithms can solve the TS search problem without knowledge of the second minimum as required by PI approaches. It further makes the problem formulation of the more general TS search problem independent of knowledge of the gradient.

Essential to the problem formulation presented is that the reaction pathway will always follow the path of the shallowest ascent and points on the path will have the lowest potential energy of all conformations with the same distance from minimum 1. Furthermore, the potential energy is subject to a steady increase along the reaction path [17]. This allows defining the reaction path as a Pareto-front in a multi-objective optimisation problem (Equation 1) that minimises potential energy and maximises distance. In particular, it allows solving the more generic TS search problem requiring only *a priori* knowledge of the first minimum, i.e. the reactant. Based on the reaction pathway the approximate TS can then be identified in a second step. The metric for the first objective, molecular potential energy, is derived from solutions of the electronic Schrödinger equation [36] using quantum chemistry software (Gaussian 03) [18]. The details of this computation are beyond the scope of this paper. The metric for the second objective is the distance between the conformation of a trial solution and that of the original reactant, "minimum 1".

A common definition for the distance between two molecular conformations with identical atoms is the least root mean square deviation (IRMSD). The IRMSD can be calculated by shifting each conformation to its geometrical centroid, finding its optimal alignment and calculating the root mean square deviation (RMSD), defined as

$$RMSD(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{N_{atoms}} \sum_{i=1}^N |\mathbf{x}_i - \mathbf{y}_i|^2} \quad (5)$$

for corresponding sets of atoms  $\mathbf{x}$  and  $\mathbf{y}$  of magnitude  $N_{atoms}$ . Each atom is represented by its Cartesian coordinates. The optimal alignment of the molecules can be efficiently computed with either the Kabsch algorithm [28, 27] or using quaternions [7].

Defining distance as a scalar value, such as IRMSD, practically allows quantifying all combined atom movements in a way meaningful to the problem domain. However, it does not provide any insight into the direction of the movement, i.e. whether it is towards or away from the TS. Furthermore, for any distance an infinite number of conformations with different (or possibly identical) potential energies can be constructed. While this does not affect the definition of the reaction path as Pareto-front as such, it poses a challenge to algorithms trying to converge towards the Pareto-front by recombining individuals from its population. To ensure convergence, several counter-measures were developed as part of this research, using the specific knowledge available in TS search. Rather than random initialisation of parameters, the initial population is generated by conducting a Monte-Carlo inspired search around the minimum. During this stage, a set of candidate solutions is created by applying small random movements to some atoms of minimum 1. To ensure sufficient diversity in the population, a larger than required set of candidates is generated and its fittest members, by energy and distance are chosen. Candidate solutions  $c$  for which

$$RMSD(c, nn_i) \gg \|RMSD(c, min1) - RMSD(nn_i, min1)\|, i \in [1, 2] \quad (6)$$

with  $nn_1$  and  $nn_2$  being the nearest neighbours as seen from minimum 1, are rejected from the approximate front. This further ensures fast convergence towards the reaction pathway by removing solutions likely to be on a different pathway. In particular for symmetric or partly symmetric molecules, it further proved beneficial to align candidate molecules to each other at regular intervals but particularly at the initialisation stage. This is to ensure that identical but rotated conformations do not slow down the convergence of the algorithm. Since rotations are calculated as part of determining the IRMSD, this process does not incur any additional computational costs and can therefore easily be performed in each iteration.

The augmented, multi-objective algorithm is essentially capable of climbing up the reaction path towards the TS, since the reaction path corresponds with the Pareto front while the search is concentrated between minimum 1 and the TS. In the region of the TS, the search must be modified to successfully identify the TS. The algorithm first needs to detect when it has reached the region of the TS and restrict further expansion of the front accordingly. One approach is to test candidate conformations for a norm of gradient close to zero [17]. However, this approach is successful only with solutions very close to the TS and the diversity inherent in the MOPSO swarm means many particles may be beyond the TS before this test can confirm its location. This may lead to potential solutions being prematurely removed from the archive. Instead, particles can detect when they are near or beyond the TS by observing changes in gradient direction, which generally points coarsely towards the nearest local minimum. If the direction of the gradient changes by angle  $\alpha \geq 90^\circ$ , then it

can be assumed that the gradient is pointing towards a new local minimum. The algorithm prevents such solutions entering the archive and corrective measures can be taken to pull solutions back into the feasible region. Refinement of solutions close to the TS then usually leads to a candidate TS within very few iterations.

A steepest-descent search (or comparable algorithm) starting from the conformation with gradient pointing towards the second minimum will quickly discover the second half of the reaction pathway. Some refinements to this method will have to be made as it occasionally fails and terminates the search prematurely. While included in the current algorithm, this feature was not used for the evaluation discussed in Section 4.

A final consideration in defining TS approximation as a multi-objective optimisation problem is an appropriate coordinate system for the parameter space. A range of alternative coordinate systems exist to describe molecules. Apart from defining the location of individual atoms by their Cartesian coordinates, some coordinate systems are based on bond lengths, valence angles etc. Such non-Cartesian coordinate systems are particularly relevant for macro-molecules, where the “curse of dimensionality” makes the use of Cartesian coordinates infeasible, as  $3 \cdot N_{Atoms}$  coordinates are necessary to describe a molecule. For small molecules, as are common in the TS search literature, Cartesian coordinates provide a suitable coordinate system for algorithmic analyses of reaction pathways. For the purposes of evaluating the proposed method, molecules with between three and seven atoms were chosen.

As the algorithm described in this paper is, at its current stage, mainly targeted at small molecules, Cartesian coordinates are the natural choice to represent the parameter space. Cartesian coordinates provide a strong coupling to the potential energy surface and allow analysing a wide range of molecules easily and without any adaptation [35]. This allows definition of the feasible space of Equation 1 as  $\mathcal{S} \subset \mathbb{R}^{3N_{Atoms}}$ . However, in principle the algorithm described in this paper could be adapted to other coordinate systems with only minor adjustments to the algorithm’s configurable parameters.

In the following Section MOPSO-specific aspects of the proposed TS search approach will be discussed.

### 3 Application of MOPSO

As outlined in Section 1, a MOPSO algorithm comprises a set number of particles that remains constant throughout the optimisation process. Particles normally start at random positions in parameter space and are allowed to move in parameter space based on a velocity equation. This velocity is based on the particle’s previous velocity, a selected globally non-dominated solution and a solution from the particle’s memory of non-dominated solutions. Each of these is weighted with a constant and a random factor, generally aiming to over- or undershoot the respective component with about equal probability. In each iteration, newly found non-dominated solutions are added to the archive, solutions no longer non-dominated are removed. By

pulling the particles towards well performing parameter vectors, the swarm usually converges towards the Pareto-front.

However, the nature of the TS search problem renders attempts to naïvely apply MOPSO or the most common multi-objective genetic algorithm, NSGA-2 [8], unsuccessful. Even given a starting population within a small range around minimum 1, neither method could converge to the minimum energy pathway within a reasonable number of iterations. Therefore, based on the considerations discussed in Section 2, a number of changes were made to MOPSO to allow it to move along the reaction path and identify the TS. These modifications comprise a Monte-Carlo inspired initialisation of the population, an adjusted guide particle selection mechanism, an adapted velocity update and the inclusion of the available knowledge of the gradient for each conformation.

To create a viable starting population a set of candidate conformations is created based on the known minimum 1. The size of this set is chosen as twice the number of particles in the swarm. Candidate conformations are created by adding a random movement to the position of the atoms of minimum 1. With each candidate created the magnitude of random dispersion increases slightly, leading to a set of candidate particles with a range of distances. From these particles the best half, measured by non-dominance and, if not sufficient for the swarm size, proximity to non-dominated solutions, is chosen as the initial population. Factors are chosen sufficiently small to keep the population within a range significantly smaller than the anticipated distance to the TS.

With the initial population within a small distance from minimum 1, the swarm will have to move up the reaction path and thereby trade off convergence towards the actual path, i.e. the Pareto-front, and progress towards the TS. Particularly with the inclusion of a gradient component, which will be discussed below, it is necessary to ensure that the swarm does not collapse back towards the minimum. This can be achieved by employing an appropriate selection mechanism for the global best guide. A wide range of guide particle selection mechanisms exist [16, 33, 6, 30]. However, all these approaches target algorithms approaching the Pareto-front top down and aim at a good diversity and spread of the solutions. The algorithm presented is designed to move up the reaction pathway, requiring a swarm pushing forward while at the same time attempting not to diverge away from the front. Adaptive grid and crowding distance based methods were thereby less successful in terms of convergence speed than roulette wheel selection of archived solutions with a strong bias towards solutions in the foremost third of the Pareto-front by distance. The reason for this is that the initial population is already reasonably close to a small section of the Pareto-front. From there it needs to follow the Pareto-front towards the TS. Adaptive grid and crowding distance based selection mechanisms are highly successful at achieving good diversity and spread across the entire Pareto-front, when the swarm moves top down towards it. This particular variation of MOPSO, however, requires a swarm that moves along the Pareto-front without diverging from it but with a constant pressure to move forward. Roulette wheel selection provides a good mechanism to achieve this pressure without moving so fast that the swarm

would diverge from the front. For the purposes of this paper non-dominated solutions not in the foremost third of the front were chosen with a probability of 10 percent.

The size of the archive is limited to 40 solutions. If this size is exceeded, solutions with the highest crowding distance are removed. However, the crowding distance for solutions in the foremost third is biased with a factor of two, effectively allowing denser crowding for particles with large distances.

An interesting aspect of the TS search problem is the availability of the gradient. While a numerical approximation of the Hessian is infeasible due to its inherent computational complexity, incorporation of the gradient allows performing correcting steps to move particles and archived solutions closer to the Pareto-front. To achieve this, the velocity update equation was amended with a gradient-based term, resulting in the following velocity equation:

$$\mathbf{v}_{t+1} = w_t \cdot \mathbf{v}_t + c_1 r_1 (\hat{\mathbf{y}}_{\text{pbest}} - \mathbf{x}_t) + c_2 r_2 (\hat{\mathbf{y}}_{\text{gbest}} - \mathbf{x}_t) + c_3 r_3 \mathbf{g}_{\text{comp},t} \quad (7)$$

where  $\mathbf{v}_t$  represents the previous velocity (momentum),  $\hat{\mathbf{y}}_{\text{pbest}}$  and  $\hat{\mathbf{y}}_{\text{gbest}}$  denote the personal and global best respectively and  $\mathbf{x}_t$  stands for the particle's parameter vector at the current iteration  $t$ . The factors  $w_t$ ,  $c_{1,2,3}$  and  $r_{1,2,3}$  represent the constant and random weights. The gradient based component is represented by  $\mathbf{g}_{\text{comp}}$ . It is defined as

$$\mathbf{g}_{\text{comp},t} = \hat{\mathbf{x}}_t - \frac{\hat{\mathbf{x}}_t \cdot (\mathbf{x}_t - \mathbf{min}_1)}{|\mathbf{x}_t - \mathbf{min}_1|^2} \cdot (\mathbf{x}_t - \mathbf{min}_1) \quad (8)$$

and serves the purpose of pushing particles towards the Pareto-front while minimizing the gradient's drag towards the minimum. An additional normalisation step is added should the magnitude of the  $\mathbf{g}_{\text{comp}}$  become greater than the magnitude of the combined other velocity components. The total velocity is furthermore subjected to clamping [14], scaling all velocities that exceed a set magnitude to a predefined maximum velocity. This step proved to improve convergence towards the TS by keeping particles close to the Pareto-front. A value close to the maximum value for random movements in the initialisation step is chosen as the maximum velocity.

In combination, these modifications allow the swarm to quickly move up the reaction path without diverging too far from it. However, streamlining the swarm in this way bears the risk of losing diversity in the population. Such a loss in diversity can significantly slow down the progress of the search. To counteract that, a turbulence component [29, 40] was added which, with a certain probability, adds a random mutation to particles. The probability was chosen as  $\frac{2}{\text{dim}\mathbf{x}}$ , where  $\text{dim}\mathbf{x}$  represents the number of parameters, selecting on average two components of the parameter vector for mutation. The mutation for each selected component is uniformly randomly chosen from the interval  $[-v_{\text{max}}, v_{\text{max}}]$ .

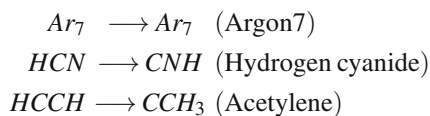
A final optional step was the possibility for archived solutions to perform a small number of gradient descent steps. When this option is included, archived solutions perform a small number of gradient descent steps (usually around 3), after each of which the new candidate solution is added to the archive iff non-dominated. To minimise the computational costs, this step is only performed every few iterations

and only on solutions that have either been newly added to the archive or where all the gradient descent steps in the previous round were successfully added to the archive. Doing these additional gradient descent steps only marginally improves the convergence and quality of the TS approximation. However, it improves the quality of the overall pathway found as part of the search.

In summary, MOPSO can be adapted to find an approximation to a chemical reaction pathway that leads up to a TS. Modifications are necessary to ensure that MOPSO stays in an area where the MOO problem definition is applicable. Primarily this is achieved by using available knowledge of the reactant to generate the initial population and applying clamping to ensure that the movements of the swarm remain relatively small. To improve the performance of the swarms movement along the path, a guide particle selection mechanism with a bias for solutions further away from the reactant is used. Other performance related changes include the incorporation of the gradient to improve convergence towards the minimum energy path and a turbulence component to improve diversity within the swarm while operating under these constraint conditions. The following section will discuss the specific parameters chosen for the algorithm as well as present the results of a number of test runs on a variety of molecules.

## 4 Results

For the evaluation of the method, three molecules with qualitatively different reaction pathways were chosen. The specific molecules chosen, the number of test runs and the metrics reported were chosen to allow comparison with related literature [17]. At the current stage most emphasis was put on the development of a suitable problem definition and MOPSO-based algorithm. Plans for qualitative comparisons with other algorithms as well as future work on making the approach more competitive will be discussed in Section 5. The following molecules were chosen for the evaluation:

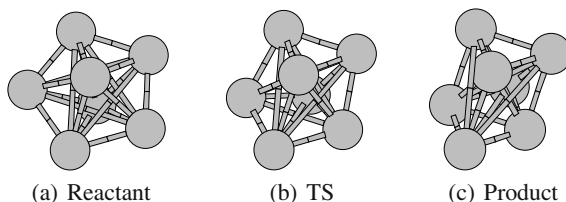


The argon 7 cluster is a well understood molecule often used for benchmarking. As a noble gas, the potential energy and gradient of argon clusters can be calculated quite cheaply using the Lennard-Jones potential [26]. The Lennard-Jones potential is applicable to all molecules comprising only neutral atoms and provides a mean of accurately approximating the potential energy of a given conformation at negligible computational costs. The potential energy of the other molecules in this evaluation was calculated using Gaussian 03 [18]. Gaussian is commercial software for computational quantum chemistry. Like the Lennard-Jones potential it allows calculating the potential energy and the gradient for a given molecular conformation, but at

substantially higher computational costs. The distance objective for all molecules was calculated as outlined in Section 2. The results from all test runs are summarised in Table 1. Shown are the actual distance between minimum and TS, the difference in potential energy between the solution found and the actual TS as well as the differences in distance, measured by distance of the candidate to the TS and by the difference in distance of the TS and minimum and candidate and minimum. Following other publications [17], the energy is measured in *kcal/mol*, distances in angstrom ( $\text{\AA}$ ). In general, the solutions achieved are quite competitive with respect to their potential energy and in many cases also with respect to their distance to the actual TS. The following paragraphs will briefly discuss the algorithm's behaviour on the different molecules.

**Argon 7** – The argon 7 cluster starts with 5 of its atoms in a regular pentagon and the two remaining atoms above and below the centre of the plane spanned by the pentagon (Figure 2(a)). Towards the TS one of the pentagon atoms (due to the symmetry this can be any one of the five) moves away from the centre while the other pentagon atoms move closer towards each other, making the pentagon irregular and giving it a pointy shape. The two atoms above and below the centre take a slightly angled position towards each other (Figure 2(b)). Towards the second minimum the pentagon shape is broken up altogether and the atoms take the shape illustrated in Figure 2(c).

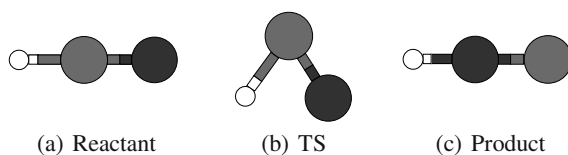
The symmetry of the molecule poses challenges to the algorithm until solutions align. On argon 7 the front will occasionally move away from the actual reaction pathway but usually converge back to its vicinity quickly. However, in one of the runs the algorithm followed a different reaction coordinate ending up with a path at a significant distance from the desired path (see table 1 run 3 of argon 7).



**Fig. 2** Minima and TS of Argon 7 ( $Ar_7$ )

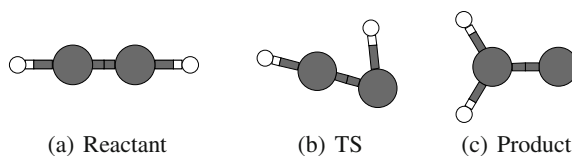
**Hydrogen cyanide** – Hydrogen cyanide starts with all three atoms in a line. Towards the transition state, Hydrogen and Nitrogen rotate around the carbon atom towards each other, forming a triangle and a bond forms between the hydrogen and nitrogen atoms. As the molecular structure approaches the second minimum, the bond between the hydrogen and carbon atoms breaks and the molecule straightens out, forming a straight line again. The process is illustrated in Figure 3. Test runs on *HCN* generally achieved good performance. An approximate transition state could be acquired in usually less than 15 iterations and with a difference in energy of only 0.001605675 *kcal/mol* on average.





**Fig. 3** Minima and TS of Hydrogen cyanide (*HCN*)

**Acetylene** – At minimum 1 of acetylene all four atoms are aligned in a straight line with the two carbon atoms in the middle. Towards the TS one of the hydrogen atoms rotates around the carbon atom it is bonded with, ending up in a position forming a triangle with the two carbon atoms. Towards the second minimum its bond changes over to the other carbon atom, eventually ending up in a Y-shape with both hydrogen atoms bonded with one of the carbon atoms. The three relevant states are illustrated in Figure 4. Acetylene proved challenging in the tests performed. The algorithm could successfully perform the rotation of the hydrogen but had a strong tendency to rotate both hydrogen atoms but one of them usually to a lesser degree. While the resulting solutions differ from the actual TS due to the second hydrogen atom, they could successfully perform the rotation and find a solution with only a small difference in potential energy to the TS. In the four test runs MOPSO successfully found one solution very close to the actual transition state and with the second hydrogen in the correct position. The three other runs found solutions with a similar potential energy but a significant distance away from the TS. Empirically this can be explained with the relatively small increase in potential energy arising from the second rotation.



**Fig. 4** Minima and TS of Acetylene (*HCCH*)

In summary, the algorithm could successfully navigate the potential energy surface, in most cases coming up with solutions close to the TS. Currently the algorithm may not reliably determine the transition reliably without the help of a human decision maker. However, user interaction could also be a pathway to significant improvements to performance. Incorporating domain knowledge would allow dynamically constraining the search space and reducing the dimensionality of the problem. For example, both *HCN* and *HCCH* could easily be constrained to only planar movements, reducing the search space by one third. Future developments will therefore consist of a combination of improvements to the algorithm itself and the incorporation of domain knowledge through user interaction. In combination, it is hoped this will allow discovery of TS on significantly larger molecules.

**Table 1** Overview of results achieved with the proposed algorithm on a range of molecules. A population of 40 particles was used for each run

Molecule	Dist $min1 \rightarrow ts$	$\delta E_a(kcal/mol)$	$Dist_{ts}(\text{\AA})$	$\delta Dist_{min}(\text{\AA})$	Iterations
<i>Ar</i> <sub>7</sub>	0.1911537	0.076588	0.018438	0.006651	48
<i>Ar</i> <sub>7</sub>	0.1911537	0.028173	0.012577	0.000053	73
<i>Ar</i> <sub>7</sub>	0.1911537	0.245090	0.117432	0.014742	89
<i>Ar</i> <sub>7</sub>	0.1911537	0.039635	0.028680	0.008752	88
<i>HCN</i>	0.3666355	0.001007	0.005780	0.008611	22
<i>HCN</i>	0.3666355	0.001568	0.013504	0.003783	15
<i>HCN</i>	0.3666355	0.001341	0.031942	0.032877	16
<i>HCN</i>	0.3666355	0.001813	0.015113	0.032877	10
<i>HCCH</i>	0.463125	0.091507	0.639619	0.445840	10
<i>HCCH</i>	0.463125	0.002149	0.311172	0.083044	23
<i>HCCH</i>	0.463125	0.013743	0.204976	0.115168	31
<i>HCCH</i>	0.463125	0.000763	0.015012	0.121488	31

## 5 Conclusion

This paper presented a novel formulation of the Transition Search (TS) search problem by treating it as a multi-objective optimisation problem and introduced a proof of concept MOPSO-based algorithm to solve this problem. At the current stage, a reasonable success rate on a range of molecules can be reported. In terms of performance, the proof of concept MOPSO algorithm still requires more objective function evaluations to find the TS than some of the other methods. This is partially alleviated by the inherent parallelism of the approach so that a comparison in wall clock time, parallel execution of the objective function assumed, is already performing similarly to some other algorithms. Some future work will be dedicated to improving the success rate and stability of the search as well as the computational costs. In particular, the incorporation of domain knowledge could greatly aid this goal. It is hoped the incorporation of domain knowledge and dynamic constraints will enable the MOPSO based approach to become a viable alternative for larger molecules which current techniques cannot process in a satisfactory amount of time.

## References

1. Baker, J.: An algorithm for the location of transition states. *Journal of Computational Chemistry* 7(4), 385–395 (1986)
2. Burger, S.K., Liu, Y., Sarkar, U., Ayers, P.W.: Moving least-squares enhanced shepard interpolation for the fast marching and string methods. *The Journal of Chemical Physics* 130, 24103 (2009)

- Burger, S.K., Yang, W.: Quadratic string method for determining the minimum-energy path based on multiobjective optimization. *The Journal of Chemical Physics* 124(5), 54109 (2006)
- Burger, S.K., Yang, W.: Sequential quadratic programming method for determining the minimum energy path. *The Journal of Chemical Physics* 127(16), 164107 (2007)
- Cerjan, C.J., Miller, W.H.: On finding transition states. *The Journal of Chemical Physics* 75(6), 2800–2806 (1981)
- Coello, C.A.C., Lechuga, M.S.: MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, IEEE Computer Society (2002)
- Coutsias, E.A., Seok, C., Dill, K.A.: Using quaternions to calculate rmsd. *Journal of Computational Chemistry* 25(15), 1849–1857 (2004)
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
- Del Campo, J.M., Köster, A.M.: A hierarchical transition state search algorithm. *The Journal of Chemical Physics* 129, 024107 (2008)
- Dewar, M.J.S., Healy, E.F., Stewart, J.J.P.: Location of transition states in reaction mechanisms. *Journal of the Chemical Society, Faraday Transactions 2: Molecular and Chemical Physics* 80(3), 227–233 (1984)
- Dey, B., Bothwell, S., Ayers, P.: Fast marching method for calculating reactive trajectories for chemical reactions. *Journal of Mathematical Chemistry* 41, 1–25 (2007), doi:10.1007/s10910-006-9060-6
- Dey, B.K., Janicki, M.R., Ayers, P.W.: Hamilton-jacobi equation for the least-action/least-time dynamical path based on fast marching method. *The Journal of Chemical Physics* 121(14), 6667–6679 (2004)
- Weinan, E., Ren, W., Vanden-Eijnden, E.: String method for the study of rare events. *Phys. Rev. B* 66, 052301 (2002)
- Eberhart, R., Simpson, P., Dobbins, R.: *Computational intelligence PC tools*. Academic Press Professional, Inc., San Diego (1996)
- Elber, R., Karplus, M.: A method for determining reaction paths in large molecules: Application to myoglobin. *Chemical Physics Letters* 139(5), 375–380 (1987)
- Fieldsend, J., Singh, S.: A multi-objective algorithm based upon particle swarm optimization. In: *Proceedings of The UK Workshop on Computational Intelligence*, pp. 34–44 (2002)
- Fournier, R., Bulusu, S., Chen, S., Tung, J.: Using swarm intelligence for finding transition states and reaction paths. *The Journal of Chemical Physics* 135(10), 104117 (2011)
- Frisch, M.J., et al.: *Gaussian 03, Revision C.02*. Gaussian, Inc., Wallingford (2004)
- Goodrow, A., Bell, A.T., Head-Gordon, M.: Development and application of a hybrid method involving interpolation and ab initio calculations for the determination of transition states. *The Journal of Chemical Physics* 129(17), 174109 (2008)
- Goodrow, A., Bell, A.T., Head-Gordon, M.: Transition state-finding strategies for use with the growing string method. *The Journal of Chemical Physics* 130(24), 244108 (2009)
- Goudos, S.K., Zaharis, Z.D., Kampitaki, D.G., Rekanos, I.T., Hilaris, C.S.: Pareto optimal design of dual-band base station antenna arrays using multi-objective particle swarm optimization with fitness sharing. *IEEE Transactions on Magnetics* 45(3), 1522–1525 (2009)

22. Henkelman, G., Jónsson, H.: Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *The Journal of Chemical Physics* 113(22), 9978–9985 (2000)
23. Henkelman, G., Uberuaga, B.P., Jónsson, H.: A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics* 113(22), 9901–9904 (2000)
24. Janson, S., Merkle, D., Middendorf, M.: Molecular docking with multi-objective particle swarm optimization. *Applied Soft Computing* 8(1), 666–675 (2008)
25. Jin, N., Rahmat-Samii, Y.: Particle swarm optimization for antenna designs in engineering electromagnetics. *J. Artif. Evol. App.*, 9:1–9:10 (January 2008)
26. Jones, J.E.: On the determination of molecular fields. ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A* 106(738), 463–477 (1924)
27. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32(5), 922–923 (1976)
28. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 34(5), 827–828 (1978)
29. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4 (1995)
30. Li, X.: A Non-Dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) *GECCO 2003. LNCS*, vol. 2723, pp. 37–48. Springer, Heidelberg (2003)
31. Maeda, S., Ohno, K.: Global mapping of equilibrium and transition structures on potential energy surfaces by the scaled hypersphere search method: applications to ab initio surfaces of formaldehyde and propyne molecules. *The Journal of Physical Chemistry A* 109(25), 5742–5753 (2005), PMID: 16833907
32. Moore, J., Chapman, R.: Application of particle swarm to multiobjective optimization. Technical report, Department of Computer Science and Software Engineering, Auburn University (1999)
33. Mostaghim, S., Teich, J.: Strategies for finding good local guides in multi-objective particle swarm optimization. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS 2003*, pp. 26–33. IEEE Computer Society (2003)
34. Peters, B., Heyden, A., Bell, A.T., Chakraborty, A.: A growing string method for determining transition states: Comparison to the nudged elastic band and string methods. *The Journal of Chemical Physics* 120(17), 7877–7886 (2004)
35. Schlegel, H.B.: Exploring potential energy surfaces for chemical reactions: an overview of some practical methods. *Journal of Computational Chemistry* 24(12), 1514–1527 (2003)
36. Schrödinger, E.: An undulatory theory of the mechanics of atoms and molecules. *Physical Review* 28(6), 1049–1070 (1926)
37. Sharaf, A.M., El-Gammal, A.A.A.: A multi objective multi-stage particle swarm optimization mopso search scheme for power quality and loss reduction on radial distribution system. In: *International Conference on Renewable Energies and Power Quality, ICREPQ* (2009)
38. Sheppard, D., Terrell, R., Henkelman, G.: Optimization methods for finding minimum energy paths. *The Journal of Chemical Physics* 128(13), 134106 (2008)

39. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, pp. 69–73 (1998)
40. Stacey, A., Jancic, M., Grundy, I.: Particle swarm optimization with mutation. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 2, pp. 1425–1430. IEEE (2003)
41. Trygubenko, S.A., Wales, D.J.: A doubly nudged elastic band method for finding transition states. *The Journal of Chemical Physics* 120(5), 2082–2094 (2004)

# A Multi-Objective Extremal Optimisation Approach Applied to RFID Antenna Design

Pedro Gómez-Meneses, Marcus Randall, and Andrew Lewis

**Abstract.** Extremal Optimisation (EO) is a recent nature-inspired meta-heuristic whose search method is especially suitable to solve combinatorial optimisation problems. This paper presents the implementation of a multi-objective version of EO to solve the real-world Radio Frequency IDentification (RFID) antenna design problem, which must maximise efficiency and minimise resonant frequency. The approach we take produces novel modified meander line antenna designs. Another important contribution of this work is the incorporation of an inseparable fitness evaluation technique to perform the fitness evaluation of the components of solutions. This is due to the use of the NEC evaluation suite, which works as a black box process. When the results are compared with those generated by previous implementations based on Ant Colony Optimisation (ACO) and Differential Evolution (DE), it is evident that our approach is able to obtain competitive results, especially in the generation of antennas with high efficiency. These results indicate that our approach is able to perform well on this problem; however, these results can still be improved, as demonstrated through a manual local search process.

## 1 Introduction

RFID [7] is the technology used to automatically identify objects at a distance without any physical or sight contact. RFID is widely used in applications such as smart

---

Pedro Gómez-Meneses · Marcus Randall  
School of Information Technology, Bond University, QLD 4229, Australia  
e-mail: [pedgomez, mrandall}@bond.edu.au](mailto:{pedgomez, mrandall}@bond.edu.au)

Pedro Gómez-Meneses  
Departamento de Ingeniería Informática, Universidad Católica de la Santísima Concepción,  
Concepción, Chile  
e-mail: [pgomez@ucsc.cl](mailto:pgomez@ucsc.cl)

Andrew Lewis  
Institute for Integrated and Intelligent Systems, Griffith University, Queensland, Australia  
e-mail: [a.lewis@griffith.edu.au](mailto:a.lewis@griffith.edu.au)

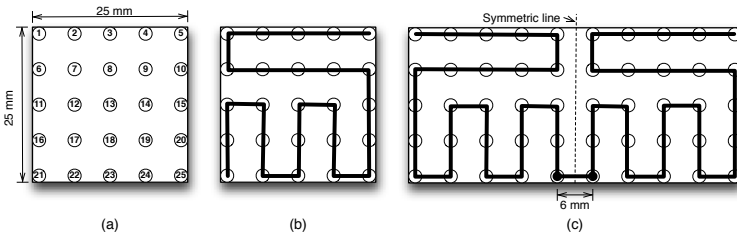
cards, electronic passports, access control, container identification, animal identification, sporting events, medical applications and industrial automation [5].

The RFID implementation requires a tag, which consists of a microchip together with a radio antenna. For a description of the principles of RFID, the reader is referred to [6, 7, 8, 12, 14]. The design of the antenna for the tag, as is demonstrated in the next section, is a Multi-Objective Combinatorial Optimisation Problem (MO-COP) [3]. Until recently, engineers primarily designed these by hand; however, nature-inspired meta-heuristics have started to be used to try to find approximated optimal configurations in the design of RFID antennas. There is an initial and isolated work with genetic algorithms [10], which was centred on a simple and regular RFID antenna design. A more systematic and general investigation has been developed on the design of meander line antennas using Ant Colony Optimisation (ACO) [8, 9, 12, 13, 14] and Differential Evolution (DE) [11]. The research with ACO and DE has demonstrated that it is possible to apply nature-inspired search mechanisms to design and evaluate meander line RFID antennas.

The above has motivated the development of an application for the design of RFID antennas for tags based on EO [1]. EO works by selecting one of the components that make up the solution manipulated by the algorithm. This selection is performed by the assessment of the contribution of each component of the solution. Hence, the component that degrades the solution evaluation the most has the highest probability of being chosen. Its value will be then replaced by a different value. Over a number of iterations, EO will gradually improve the quality of solutions it generates. The implementation of the EO approach for this problem is an interesting challenge because the evaluation for any valid solution of an antenna layout is performed by an external black-box module provided by the free antenna modelling software package, NEC [2]. In terms of the meta-heuristic, there is no information about the potential value of components with which to guide the search. Therefore, an inseparable fitness evaluation mechanism is required to be implemented. One such scheme is developed as part of this research.

The inseparable fitness evaluation technique will be based on the idea of the global pheromone scheme used in ACO. This scheme will serve to maintain a record, over generations, of the components present in the solutions that have good evaluations. Thus, components with a higher presence in good solutions will have a higher level of pheromone, which means that these components will be unlikely to change as a result of EO's selection mechanism.

The creation of a meander line is the most common form to design a RFID antenna. The antenna is a folded dipole that can be represented by a Cartesian grid [6, 12] (see Figure 1), which is symmetrical around the dipole. The grid is comprised of a finite number of points in a square shape. The meander line is created by the connection of these points by horizontal or vertical segments, so that no segment intersects another already established. It is not mandatory that all grid points be connected; however, the meander line must start on any adjacent point to



**Fig. 1** (a) The grid defines a  $5 \times 5$  antenna. (b) Illustrates a potential meander line antenna for the grid defined in (a). (c) The dipole antenna generated through the binding of two mirrored meander line antennas shown in (b) by a 6 mm join.

the symmetric line that divides the two halves of the antenna. The start point is used to connect, with an additional line, the two-mirrored dipoles of the antenna.

Recently, Randall et al. [12] developed an initial software system to design and evaluate the efficiency of meander line RFID antennas based on ACO. The approach was successfully applied on grids from size  $5 \times 5$  grid points to  $10 \times 10$  as a constructive mechanism creating a path on the grid. Continuing this research, Weis et al. [13] proposed an extension based on a local search technique that uses an operator known as backbite. The next step was developed by Lewis et al. [8, 9] who proposed a multi-objective version, which maximises the efficiency and minimises the resonant frequency. After that, Weis et al. [14] proposed a mechanism to increment the performance of the ACO software using an initial pheromone biased by the a-priori knowledge from human designers. Each one of these works show an improvement for all grid instances compared to those obtained by the previous work.

Montgomery et al. [11] developed a multi-objective DE approach, which differs from ACO in the way the solution is created. Instead of using a constructive mechanism, the DE algorithm starts with a population of complete initial solutions, which is iteratively improved using vector differences to modify it. The results showed that DE is an appropriate method to achieve good solutions, being more effective for smaller antenna grid sizes. The DE approach did not use any local search mechanism to improve the solutions. The DE approach demonstrates that it is possible to implement a non-constructive technique to solve the RFID antenna design problem, which is inherently constructive by nature. However, DE is a search method more suitable to solve continuous optimisation problems and the RFID antenna design problem has been modelled as a combinatorial problem. For this reason, it is believed that the non-constructive EO meta-heuristic, whose search method is especially suitable for solving combinatorial optimisation problems, could also achieve competitive results to those generated by the ACO techniques.

The EO approach described in this paper is tested and compared on the same grid sizes as those used in the multi-objective ACO approach of Lewis et al. [9] and the multi-objective DE approach of Montgomery et al. [11].



The rest of this paper is organised as follows. Section 3 explains the inseparable fitness evaluation scheme to solve problems that do not provide the necessary information to perform a separable evaluation of components. Section 3 presents the extremal optimisation approach applied to the RFID antenna design problem. Section 4 gives a summary of the computational experiments developed with an analysis of them. Section 5 presents the conclusions and discusses the future work arising from this study.

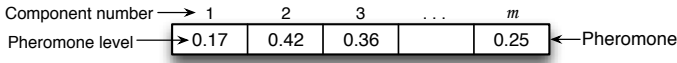
## 2 The Inseparable Fitness Evaluation Scheme

In EO, each component of the solution is evaluated to assess its contribution to the quality of the optimal. This evaluation is done at every iteration and it is essential that the problem to be solved has as much suitable information as possible for this purpose. However, not all problems can provide this information. For this reason, it is necessary to incorporate a mechanism into EO in order to solve problems where it is not possible to carry out separable component fitness evaluations.

In the RFID antenna design problem, the input data are: the grid size, the side antenna size, the wire width and the dipole segment connection length. This information, along with a proposed antenna design, is used as input to the NEC software. This software works as a black box computation, which returns the evaluation with regard to the efficiency and the resonant frequency. No other data is available or known about these calculations, which may be used to perform the component fitness evaluation.

The inseparable fitness evaluation scheme proposed herein is based on the pheromone structures used by Ant Colony System (ACS) [4]. In ACS, each time an ant traverses a path from the nest to the food source, a local pheromone update is done. This action is performed until a complete route is produced, updating the pheromone level at each section of the route. Once all the ants have finished the creation of a route from the nest to the food source, a second update is carried out. Here, the global best route found until the current iteration is used as a global pheromone update for each section in this route. As EO is not a method that generates a solution by applying a constructive mechanism, only the global pheromone update is considered.

The idea behind the use of the pheromone structure is motivated by the fact that it works as an element which can keep a record of the solution components that are present in good solutions. In this way, each time a new non-dominated solution is found, a pheromone update is performed. With this procedure, the components that degrade the solution will have low pheromone levels and the components that contribute to the optimality of the solution will have high pheromone levels. Thus, the pheromone structure allows the implementation of a fitness evaluation of the components of the solution without any specific information about the evaluation of the objective function(s).



**Fig. 2** An example of a pheromone vector for the RFID antenna design problem, where  $m$  is the number of components in the solution.  $m$  is formally defined in the next section.

In the particular case of the RFID antenna design problem, each component of the pheromone structure represents a segment that joins two adjacent non-diagonal points of the grid. The collection of these segments composes one half of the symmetrical design. Figure 2 illustrates the use of pheromone.

The pheromone vector is initialised with a very low value between zero and one. After that, the pheromone level for each component is updated along the extremal optimisation search process according to Equation 1

$$\Phi_i^t = (1 - \alpha)\Phi_i^{t-1} + \alpha\Delta \tag{1}$$

where:

- $\Phi_i^t$  is the pheromone value for the  $i^{\text{th}}$  component at time  $t$ ,
- $\alpha$  is the pheromone decay factor, and
- $\Delta$  is the pheromone positive reinforcement for the  $i^{\text{th}}$  component if it belongs to a non-dominated solution, 0 otherwise.

This pheromone update is applied only to each non-dominated solution found by EO and only the components that have allocated a segment that belongs to the solution are able to obtain this reinforcement. Finally, the inseparable fitness evaluation scheme of a solution is given by Equation 2

$$\lambda(x_i) = \Phi(x_i), \quad \forall i \quad 1 \leq i \leq n \tag{2}$$

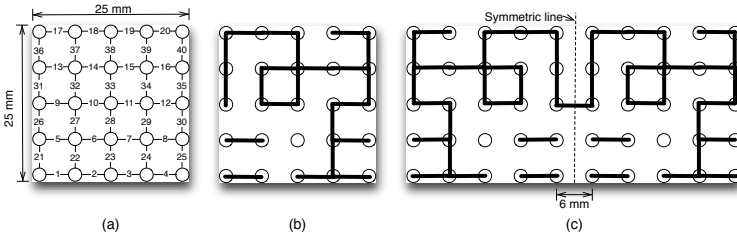
where:

- $\lambda(x_i)$  is the fitness evaluation for the  $i^{\text{th}}$  component of the solution,
- $\Phi(x_i)$  is the pheromone level for the  $i^{\text{th}}$  component of the solution,
- $n$  is the number of components of the solution.

Note that the fitness function evaluation, for each component of the solution, is completely defined by the pheromone structure. It is unnecessary to have any other information with respect to this particular problem. This scheme is very general and applicable to other black box problems.

### 3 EO Applied to RFID Antenna Design

Until now, all approaches based on either ACO or DE, have worked with a meander line antenna design. These works fulfilled the objective of demonstrating that it is possible to develop a software system to design and evaluate meander line RFID antennas using both constructive and non-constructive meta-heuristics. However, new questions emerge as to the possibility of applying other nature-inspired



**Fig. 3** (a) The grid representation for a  $5 \times 5$  antenna based on segment allocation. (b) Illustrates a potential modified meander line. (c) The dipole antenna generated with two mirrored modified meander lines connected by a 6 mm segment.

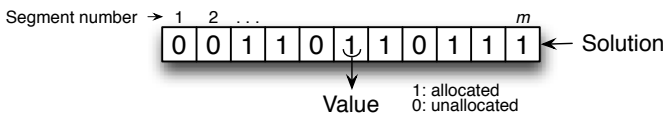
meta-heuristics for generating antenna designs and if it is possible to build efficient antennas with a different design to that of a meander line.

In response to these questions, the idea of developing an EO approach using modified meander lines, is proposed. Here, the restriction of generating a single continuous line is relaxed and now a line with loops, mesh segments and parasitic (isolated) elements can be designed. This revised scenario involves a new interpretation of how the grid is built.

In previous works, each *node* on the grid is tagged with an identification number from 1 to  $n^2$ , where  $n$  is the grid size, as was shown in Figure 1. Therefore, the solution is represented by a sequence of nodes, which are joined in order by a continuous line to generate the antenna design. However, in a modified meander line antenna design scheme, the same grid has a different interpretation. Here, the *segments* that join the nodes are tagged with an identification number from 1 to  $m$ , where  $m = (2 \times (n - 1) \times n)$ , as shown in Figure 3.

For the modified meander line case, the solution is represented by a vector of  $m$  components. Each component represents a segment that could potentially be put in the antenna design. If the  $i^{\text{th}}$  segment is allocated to the antenna design, then a value of 1 is assigned to its position in the solution vector. A value of 0 is assigned to signify that the segment is not part of the antenna design (see Figure 4).

This representation to design a general modified meander line antenna becomes a combinatorial optimisation problem similar to the knapsack problem (KP). The main difference between KP and the problem defined in this paper, is that the weight for each knapsack item is known, whereas in the modified meander line antenna design there is no information related to any sort of weight that is associated with the segments. Also, there is the constraint that a segment associated with the edge of



**Fig. 4** An example of a solution vector for the modified meander line antenna

the grid aligned to the symmetric line, must be allocated. Taking into consideration these differences, the modified meander line antenna design will be formulated as a multi-objective KP, which must maximise efficiency and minimise resonant frequency.

The initial solution is obtained by a random selection of segments. This mechanism could generate an infeasible solution due to the requirement that at least one segment associated with the edge of the grid aligned to the symmetric line must be in the antenna design. This is because the additional 6 mm segment is used to connect and energise the two-mirrored grids that form the dipole antenna. If an initial feasible solution is generated then this is added as the first point in the approximated Pareto-front set.

For this particular problem, which has no available information to perform an assessment of the components of the solution, a differentiated fitness evaluation is implemented with the support of the inseparable fitness evaluation scheme described in Section 3. Equation 3 illustrates the fitness evaluation to select the segment that degrades the solution to be assigned or unassigned from the antenna design.

$$\lambda(x_i) = \begin{cases} \Phi(x_i) , \forall i \quad 1 \leq i \leq m , \text{ for feasible solutions} \\ -V_i , \forall i \in \tilde{M} , \text{ for infeasible solutions} \end{cases} \quad (3)$$

where:

- $\Phi(x_i)$  is the pheromone level for the  $i^{\text{th}}$  component of the solution,
- $m$  is the number of components of the solution,
- $V_i$  is the number of segments connected to the  $i^{\text{th}}$  unassigned segment associated to the edge of the grid aligned to the symmetric line, and
- $\tilde{M}$  is the set of segments associated to the edge of the grid aligned to the symmetric line.

When the solution is feasible, the inseparable fitness evaluation based on the pheromone structure is applied to define which components of the solution degrade the solution. Thus, segments that have a low level of pheromone are chosen with a higher probability to be allocated/unallocated than those with a high level of pheromone. This is because a segment with a low level means that it does not contribute to achieving a high efficiency with a low resonant frequency in the modified meander line antenna design. On the other hand, when the solution is infeasible, this means that none of the segments on the symmetric line are allocated. To obtain a feasible solution again, one of them must be selected to be part of the antenna design. Hence, unallocated segments on the symmetric line, with a high number of neighbouring segments that could be connected to it, are considered to be the components that degrade the solution. This is because those unallocated segments have a higher probability of generating a longer continuous line when this is connected to a segment forming part of a long line. This is important as longer continuous lines generally reduce resonant frequency.

The Roulette Wheel Selection (RWS) scheme is used to select one of the worst evaluated segments according to its probability  $P$ . The replacement of the value for

the chosen component (segment) is a simple process because it only has to change its value from 0 to 1 or from 1 to 0, as appropriate.

Because of the fact that the NEC software frequently requires a considerable amount of time to evaluate a solution, for now, secondary search is not performed. However, a local search performed manually, to discern a possible future efficient implementation of this, is carried out. For the same reason, as in previous works [9, 8, 11, 12, 13, 14], a caching system is implemented to reduce the number of evaluations made by the NEC software and to decrease the runtime of the EO process.

Each time a feasible solution is found, the non-dominated procedure is activated to update the approximated Pareto-front set found at the current iteration. If a new non-dominated solution is found, then the pheromone level update is performed. The EO procedure is repeated for a pre-set number of iterations. Finally, the approximated Pareto-optimal and Pareto-front sets that were found by the approach are returned as output. Algorithm 1 shows the EO pseudocode for the bi-objective modified meander line antenna design problem.

---

**Algorithm 1** The EO pseudocode for the RFID antenna design problem.

---

```

1: Generate the probability vector  $P$ 
2: Generate the initial pheromone structure  $\Phi$ 
3: Initialise a solution  $S$  with random values from  $\{0, 1\}$ 
4: if  $S$  is feasible then
5:   Add  $S$  into the approximated Pareto-front set
6:   Update pheromone levels  $\Phi$ 
7: end if
8: repeat
9:   if the current solution is feasible then
10:    Evaluate the fitness  $\lambda(x_i)$  for the components of the feasible solution
11:   else
12:    Evaluate the fitness  $\lambda(x_i)$  for the components of the infeasible solution
13:   end if
14:   Rank the components according to its fitness  $\lambda(x_i)$  from the worst to the best
15:   Select a component based on the probability of its rank  $P$  using RWS
16:   Obtain a  $S_{new}$  by changing the value of the selected component in  $S$  to 0 or 1 as appropriate
17:   Evaluate the new solution  $S_{new}$ 
18:   if  $S_{new}$  is feasible then
19:     Apply the NonDominance( $S_{new}$ ) procedure
20:     if a new non-dominated solution is generated then
21:       Update pheromone levels  $\Phi$ 
22:     end if
23:   end if
24: until the termination condition is satisfied
25: return the approximated Pareto-optimal and Pareto-front sets

```

---

## 4 Computational Experiments

The EO algorithm was coded in the C language and compiled with `gcc`. The tests were performed on a Linux workstation with a 3 GHz Core2 Duo CPU and 3.8 GB of RAM. The NEC software was the version available in late October 2010.

The antenna attributes were the same as used in the previous works by Lewis et al. [8] and Montgomery et al. [11]. Tests consist of six grid sizes, from  $5 \times 5$  to  $10 \times 10$ . All grids have a dimension of  $25 \times 25$  mm and the segment width is 1 mm. An extra segment of 6 mm in length is used to connect the dipole.

The number of iterations for the search was set to 1000. Each instance was run three times using a different random seed. Grids from  $5 \times 5$  to  $7 \times 7$  ran an extra search with 2000 and 3000 iterations only to observe if lower resonant frequencies could be reached. The work at this time is exploratory and preliminary and the main aim of this research is to prove the viability of EO for RFID antenna design and observe its behaviour for further development.

The initial pheromone level and the pheromone decay factor were set at 0.1, as these values have been found to be robust by Dorigo and Gambardella [4]. The pheromone value was set at 1.

### 4.1 Comparison of Results

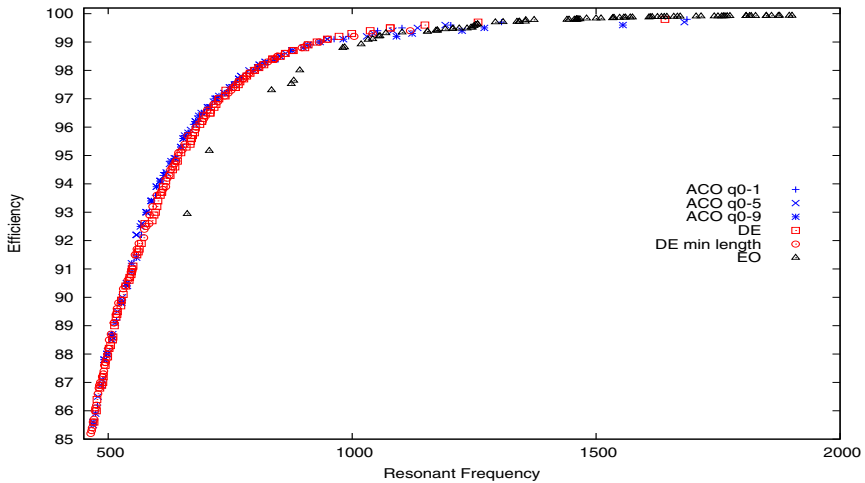
The EO results were compared with those obtained by the ACO [9, 8] and the DE [11] algorithms. These variants are denoted as  $ACO_{q1}$ ,  $ACO_{q5}$ ,  $ACO_{q9}$ ,  $DE$ ,  $DE_{minL}$ . The  $q$  index for the ACO approaches refers to three different levels of greediness to determine the next component to add in its constructive process; that is, the probability that a greedy decision is made instead of a probabilistic one. On the other hand, the DE variant  $minL$  alludes to the incorporation of a minimum length constraint in the meander line to encourage the exploration of solutions with lower resonant frequencies.

The objective of this comparative study is to discuss the strengths and weaknesses of the EO approach and the contribution that this new technique for RFID antenna design can provide. To do this, it is necessary to consider the following implementation differences that affect this comparison:

- The ACO and DE approaches work under the restriction of generating meander lines, which can have different lengths. The EO algorithm, however, develops an antenna based on the presence or absence of the segments that make up the grid through a modified meander line that is able to create any design.
- The connection line is another aspect that has been dealt with in different ways. The DE approach always places the connection line, which is the link connecting the two halves of the dipole antenna, at node 1, while the ACO and EO algorithm may make the connection at any node on the symmetrical line.
- The local search mechanism has only been implemented by the ACO approach through the application of the backbite operator [13]. The DE and EO algorithms have not yet implemented a local search mechanism.

- The number of solutions produced by DE and ACO were 10100 and 10000 solutions respectively. However, EO produced only between 1000 and 3000 solutions according to the performed experiment.
- DE and ACO, are population-based techniques. The variability of new solutions generated by population-based methods is generally greater than the single-based methods, as is the case for extremal optimisation.

Figure 5 shows the plotted graph results for the ACO, DE and EO approaches for the grid size  $7 \times 7$ . The remaining plotted graphs can be obtained from <http://it.bond.edu.au/evolve2012/>.



**Fig. 5** Results for the  $7 \times 7$  bi-objective RFID antenna design problems. This graph is indicative of the performance of EO, regardless of grid size.

The first issue that can be evident in the graph is an approximated Pareto-front set that is biased towards the objective related to efficiency. This can be interpreted as EO generating designs with a high efficiency instead of antenna designs with low resonant frequencies. This is evident by observing the number of solutions generated in the frequency range 1 GHz to 2 GHz. Furthermore, most of the solutions generated in that frequency range achieved better efficiency than those generated by the previous approaches.

Use of the S-metric [15] and the C-metric [15] can achieve a more analytical comparison than the visual analysis of the plotted graphs. The S-metric evaluates how much of the multi-objective space is dominated by a given approximated Pareto-front set. The C-metric compares two approximated Pareto-front sets by measuring the proportion of points in one set that are dominated by the other set.

From Table 1 it can be observed that EO achieved a similar coverage of the dominated space in comparison with the other approaches for the grids of size  $5 \times 5$ ,

**Table 1** The S-metric values for the RFID antenna design with a grid size of  $5 \times 5$ ,  $6 \times 6$ ,  $7 \times 7$ ,  $8 \times 8$ ,  $9 \times 9$  and  $10 \times 10$

Grid Size	$ACO_{q1}$	$ACO_{q1}$	$ACO_{q9}$	DE	$DE_{minL}$	EO
$5 \times 5$	49331.8	48658.8	48600.0	49326.1	48053.9	49422.6
$6 \times 6$	86862.4	86557.3	86081.1	86949.5	86898.8	86319.2
$7 \times 7$	48587.1	48333.5	48005.2	48583.5	47525.4	47118.1
$8 \times 8$	27738.0	27738.0	27473.7	27684.6	27201.7	19713.8
$9 \times 9$	48250.1	48177.5	47847.0	47835.6	46291.9	36255.8
$10 \times 10$	47477.0	47333.5	47303.5	46678.4	44753.3	35810.6

$6 \times 6$  and  $7 \times 7$ . However, for the grids of size  $8 \times 8$ ,  $9 \times 9$  and  $10 \times 10$ , the S-metric indicated a lower dominated space than the ACO and DE approaches. The decrease in the value of the metric for the three largest grids was due to the reduced number of iterations that EO carried out (just 1000, versus 3000 performed by the three smaller grids). This demonstrates that EO is able to find new, non-dominated solutions as the number of iterations is increased. It is important to note that despite the similarity of the S-metric value achieved by it, the approximated Pareto-front sets obtained by EO is more inclined towards higher resonant frequencies than lower ones.

The C-metric values shown in Table 2 illustrate that even with the low number of iterations performed by EO, the proportion of solutions found for this, that are dominated by the other approaches, is less than 32%. Furthermore, this became zero in the case of the  $9 \times 9$  grid. These values can be considered to be good, but it is necessary to develop a deeper analysis to understand why these values are received.

**Table 2** The C-metric values for grid sizes of  $5 \times 5$ ,  $6 \times 6$ ,  $7 \times 7$ ,  $8 \times 8$ ,  $9 \times 9$  and  $10 \times 10$

$5 \times 5$							$6 \times 6$						
$ACO_{q1}$	$ACO_{q5}$	$ACO_{q9}$	DE	$DE_{minL}$	EO		$ACO_{q1}$	$ACO_{q5}$	$ACO_{q9}$	DE	$DE_{minL}$	EO	
-	0.290	0.347	0.222	0.177	0.279		-	0.304	0.442	0.131	0.219	0.161	
0.083	-	0.152	0.044	0.044	0.102		0.146	-	0.442	0.171	0.243	0.161	
0.0	0.036	-	0.044	0.044	0.102		0.08	0.130	-	0.105	0.134	0.147	
0.027	0.2	0.173	-	0.0	0.205		0.333	0.391	0.519	-	0.207	0.279	
0.013	0.163	0.152	0.044	-	0.205		0.293	0.391	0.480	0.157	-	0.25	
0.027	0.109	0.065	0.066	0.022	-		0.04	0.072	0.038	0.013	0.024	-	
$7 \times 7$							$8 \times 8$						
$ACO_{q1}$	$ACO_{q5}$	$ACO_{q9}$	DE	$DE_{minL}$	EO		$ACO_{q1}$	$ACO_{q5}$	$ACO_{q9}$	DE	$DE_{minL}$	EO	
-	0.285	0.229	0.559	0.397	0.301		-	0.0	0.517	0.689	0.656	0.066	
0.245	-	0.208	0.623	0.479	0.290		0.0	-	0.517	0.689	0.656	0.066	
0.210	0.116	-	0.537	0.397	0.107		0.173	0.173	-	0.655	0.585	0.016	
0.333	0.181	0.187	-	0.173	0.311		0.269	0.269	0.344	-	0.474	0.083	
0.315	0.233	0.25	0.731	-	0.161		0.057	0.057	0.241	0.333	-	0.016	
0.017	0.025	0.104	0.010	0.0	-		0.038	0.038	0.137	0.011	0.0	-	
$9 \times 9$							$10 \times 10$						
$ACO_{q1}$	$ACO_{q5}$	$ACO_{q9}$	DE	$DE_{minL}$	EO		$ACO_{q1}$	$ACO_{q5}$	$ACO_{q9}$	DE	$DE_{minL}$	EO	
-	0.573	0.608	0.824	0.982	0.051		-	0.408	0.701	0.886	1.0	0.112	
0.166	-	0.540	0.72	0.965	0.038		0.39	-	0.656	0.806	1.0	0.105	
0.102	0.16	-	0.662	0.844	0.0		0.123	0.098	-	0.647	0.95	0.090	
0.166	0.253	0.297	-	0.87	0.102		0.109	0.197	0.388	-	1.0	0.127	
0.012	0.04	0.148	0.094	-	0.0		0.0	0.0	0.014	0.0	-	0.052	
0.025	0.013	0.094	0.0	0.017	-		0.013	0.014	0.029	0.011	0.033	-	



A possible reason for these results is because of the fact that EO creates a type of antenna with different features to those generated by the previous works.

## 4.2 *Manual Local Search Analysis*

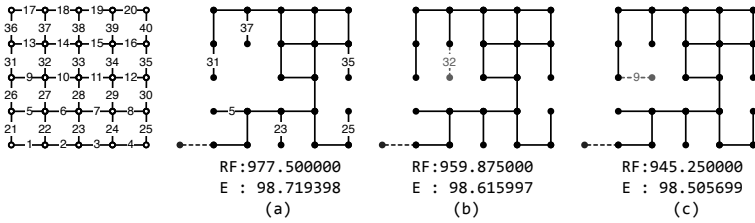
Previous work leaned towards the creation of antennas through a continuous line that follows the pattern of a meander path. Generally, the longer the continuous line, the lower the resonant frequency and quicker the NEC evaluation. However, the EO implementation investigates the free allocation of segments in the grid generating antenna designs that do not necessarily follow any predetermined pattern. Therefore, the probability of creating a continuous long line is lower. This feature has a strong impact on both increasing the evaluation runtime for the NEC software and decreasing the generation of antennas with lower resonant frequencies.

Results showed that EO was able to generate similar approximated Pareto-front sets for runs with different initial seeds. Also, when the iterations are increased, the approach is able to find points on the attainment surface with better convergence and diversity. However, this improvement was not expected to reach solutions with lower resonant frequencies.

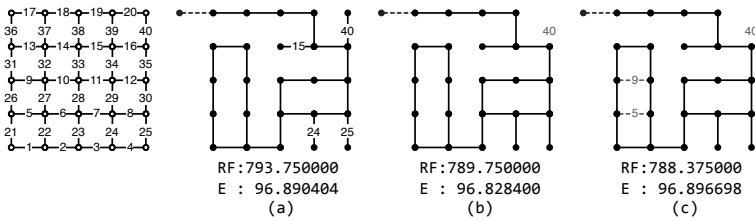
To improve the convergence towards lower resonant frequencies, a potential local search mechanism was analysed. The backbite local search mechanism used in previous works cannot be applied to the EO approach as this technique is oriented to the construction of Hamiltonian walks as in the case of a meander line. However, EO needs a mechanism to generate, in the current modified meander line solution, a continuous line as long as possible through the modification of one or more segments. The difficulty of performing this seemingly simple operation lies in the modified meander line generated by EO not following any predefined pattern such as a meander line (see Figure 3). The drawn line on the grid can simultaneously produce continuous lines, branched lines, circuits and completely isolated lines.

A heuristic that can be followed is to identify all end segment lines and to try to add a new segment without creating a circuit. For example, Figure 6(a) shows the minimum resonant frequency (RF) with its associated efficiency (E) that was found in the  $5 \times 5$  grid. Here, all end segment lines that potentially can add an adjacent segment are identified. From these points, only segments 31 and 37 are able to add a new segment without generating a circuit. Figure 6(b) shows the result when segment 32 is incorporated. It can be seen that the resonant frequency decreases, but also there is a decrease in efficiency. A better resonant frequency is achieved when segment 9 is added (see Figure 6(c)). This improvement is because the incorporation of segment 9 achieves a longer continuous line.

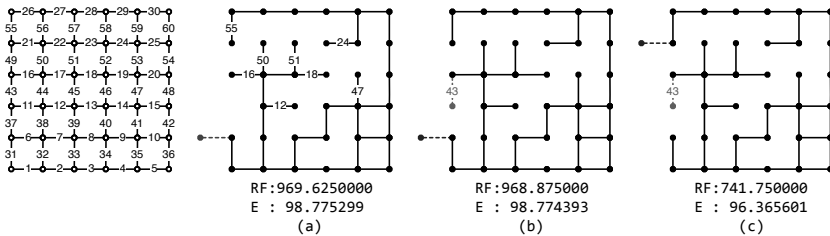
A special case can be given in Figure 7(a), which illustrates when it is not possible to add an adjacent segment without generating a circuit. Here, there is the option of removing any of the branched segment lines as can be seen in Figure 7(b). The elimination of one of these branched segment lines achieved a fall in the resonant frequency. This is an interesting finding that requires more analysis; however, this was not the only discovery. It is possible to reduce the resonant frequency even



**Fig. 6** A manual local search case for the  $5 \times 5$  grid, when it is possible to add a new segment without generating a circuit. The dashed segment represents the connection line.



**Fig. 7** A manual local search case for the  $5 \times 5$  grid, when it is not possible to add a new segment without generating a circuit. The dashed segment represents the connection line.



**Fig. 8** A manual local search case for the  $6 \times 6$  grid, given a change to the connection line to a different segment in the symmetrical line side. The dashed segment represents the connection line.

more with a slight increase in efficiency, which is desirable in RFID. This is possible when segment 40 is eliminated and segments 5 and 9 are incorporated into the circuit that is generated at the end of the continuous segment starting from the line of connection (see Figure 7(c)). For these two last cases, it is necessary to create a general local search heuristic that allows for the discovery of new non-dominated points towards lower resonant frequencies.

A third interesting case to discuss is given by analysing the solution with the lowest resonant frequency obtained by EO for the grid size  $6 \times 6$ . Figure 8(a) shows the antenna design and identifies the end segment lines that potentially could add an adjacent segment. From these segments, the only one that can add a new segment without generating a circuit is segment 16. When segment 43 is added to segment

16, the resonant frequency has a slight decrease from 969.625 MHz to 968.875 MHz (see Figure 8 (b)). However, if the connection line is considered in the local search process, the simple act of changing the connection point to a new position could achieve a significant improvement in the reduction of the resonant frequency. Figure 8 (c) shows the considerable fall in the resonant frequency from 968.875 MHz to 741.75 MHz when the connection line is moved from segment 21 to segment 55.

The incorporation of a local search mechanism is a task that requires some time to be studied. The aim is to develop a mechanism that requires as few assessments as possible due to the considerable amount of time required for the NEC software. This is because the local search could multiply the number of solutions generated to be evaluated by the NEC software by a factor determined by how many additional solutions are generated by it.

## 5 Conclusions

This paper described the implementation of an EO algorithm, which was applied to solve a real-world RFID antenna design problem.

The need to extend fitness evaluation to deal with problems that have no available information or data to perform the fitness evaluation for the components of the solution was evident. In response to this requirement, the inseparable fitness evaluation technique was proposed. The implementation of this technique used a pheromone-based mechanism (borrowed from ACO) to assess the contribution of the components of the solution.

A new representation of the solution based on the allocation of segment lines was implemented. With this novel representation, the RFID antenna design was set out as a KP, which allowed the generation of a new type of antenna design through a modified meander line.

Results show that the EO is an effective and efficient mechanism to find antenna designs with high efficiency for resonant frequencies over 1 GHz. The new scheme of the modified meander lines generated by EO has features that the traditional meander line does not, such as an increase in the antenna inductive load. However, it is believed that these same features prevent the new approach from achieving low resonant frequency designs, which are efficiently produced by the ACO algorithms. For this reason, a set of mechanisms, to be implemented as extensions to the EO approach, improving its convergence towards antenna design with low resonant frequencies, were proposed.

Despite the low number of evaluations that were performed with the EO approach (between 1000 and 3000 evaluations, depending on the size of the instance) compared with the 10000 evaluations performed by previous works, the preliminary results are very promising. A number of potential future works emerge from the proposal. Firstly, a population-based extension of the EO approach is an interesting aspect to be developed, with the aim of increasing the diversity of the solutions. This improvement will be the basis for a parallel implementation of the EO

algorithm, with the objective of reducing the considerable wall-clock time required by the NEC evaluation software. Finally, taking into consideration the good performance achieved by EO to find antenna designs with high efficiencies and by ACO to find antenna designs with low resonant frequencies, a new hybrid heuristic that uses both methods could be implemented. It is believed that with this new hybrid heuristic, enhanced approximated Pareto-front sets could be found.

**Acknowledgements.** The first author is supported by a Bond University Publication Scholarship.

## References

1. Boettcher, S., Percus, A.G.: Extremal optimization: methods derived from co-evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 825–832 (1999)
2. Burke, G., Poggio, A., Logan, J., Rockway, J.: NEC - numerical electromagnetics code for antennas and scattering. In: Antennas and Propagation Society International Symposium, vol. 17, pp. 147–150 (1979)
3. Coello Coello, C.A., Dhaenens, C., Jourdan, L.: Multi-Objective Combinatorial Optimization: Problematic and Context. In: Coello Coello, C.A., Dhaenens, C., Jourdan, L. (eds.) *Advances in Multi-Objective Nature Inspired Computing*, SCI, vol. 272, pp. 1–21. Springer, Heidelberg (2010)
4. Dorigo, M., Gambardella, L.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1(1), 53–66 (1997)
5. Finkenzeller, K.: *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*, 3rd edn. John Wiley & Sons (2010)
6. Galehdar, A., Thiel, D., O’Keefe, S., Kingsley, S.: Efficiency variations in electrically small, meander line RFID antennas. In: Antennas and Propagation Society International Symposium, pp. 2273–2276. IEEE (2007)
7. Landt, J.: The history of RFID. *IEEE Potentials* 24(4), 8–11 (2005)
8. Lewis, A., Randall, M., Galehdar, A., Thiel, D., Weis, G.: Using Ant Colony Optimisation to Construct Meander-Line RFID Antennas. In: Lewis, A., Mostaghim, S., Randall, M. (eds.) *Biologically-Inspired Optimisation Methods*. SCI, vol. 210, pp. 189–217. Springer, Heidelberg (2009)
9. Lewis, A., Weis, G., Randall, M., Galehdar, A., Thiel, D.: Optimising efficiency and gain of small meander line RFID antennas using ant colony system. In: Proceedings of the 11th Congress on Evolutionary Computation, pp. 1486–1492. IEEE Press (2009)
10. Marrocco, G.: Gain-optimized self-resonant meander line antennas for RFID applications. *IEEE Antennas and Wireless Propagation Letters* 2(1), 302–305 (2003)
11. Montgomery, J., Randall, M., Lewis, A.: Differential evolution for RFID antenna design: a comparison with ant colony optimisation. In: Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, pp. 673–680. ACM (2011)
12. Randall, M., Lewis, A., Galehdar, A., Thiel, D.: Using ant colony optimisation to improve the efficiency of small meander line RFID antennas. In: Proceedings of the 3rd International Conference on e-Science and Grid Computing, pp. 345–351. IEEE Computer Society (2007)

13. Weis, G., Lewis, A., Randall, M., Galehdar, A., Thiel, D.: Local search for ant colony system to improve the efficiency of small meander line RFID antennas. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, pp. 1708–1713. IEEE (2008)
14. Weis, G., Lewis, A., Randall, M., Thiel, D.: Pheromone pre-seeding for the construction of RFID antenna structures using ACO. In: Proceedings of the 6th International Conference on e-Science, pp. 161–167. IEEE Computer Society, Brisbane (2010)
15. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: methods and applications. Ph.D. thesis, Swiss Federal Institute of Technology, ETH (1999)

# Wasp Colony with a Multiobjective Local Optimizer for Dynamic Task Planning in a Production Plant

Luis Fernando Gutierrez-Marfileno, Eunice Ponce-de-Leon, Elva Diaz-Diaz, and Leoncio Ibarra-Martinez

**Abstract.** Dynamic task scheduling is a time-dependent optimization issue. In this work, we modeled the process that is performed at a production plant as a task scheduling issue, in which a production line sends trucks to a painting plant with several stations. The objective is to attain efficient task scheduling, taking into account three conflicting objectives: number of color changes in booths, work tardiness, and makespan. In order to solve this problem, we developed a hybrid technique, which comprises a Wasp Colony algorithm and a set of priority rules. Both the problem and its solution were modeled through Agent Unified Modeling Language (AUML) so as to achieve implementation. The results were a remarkable decrease in the number of color changes and work tardiness and the preservation of the number of painted trucks within an acceptable magnitude.

## 1 Introduction

Daily task scheduling is a time-dependent optimization problem and, where workload is not stationary and the environment is changing (as tasks are dynamically created or destroyed or they appear in bursts), task scheduling should be performed in real time and based on task attributes; the needs of future tasks are not known *a priori*. This model comprises a set of specific resources (individuals, processors, machines, etc.) that are allocated to specific tasks [24]. One application of this problem is machine scheduling for the performance of manufacturing tasks in manufacturing companies [7].

Dynamic task scheduling algorithms require high performance when facing the problem's dynamic nature during the execution of production plans in the manufacturing shops and its goal is to optimize one or more objective functions [24].

---

Luis Fernando Gutierrez-Marfileno · Eunice Ponce-de-Leon · Elva Diaz-Diaz · Leoncio Ibarra-Martinez

Universidad Autonoma de Aguascalientes

e-mail: [lfgutie@correo.uaa.mx](mailto:lfgutie@correo.uaa.mx), [eponce@correo.uaa.mx](mailto:eponce@correo.uaa.mx), [ediazd@correo.uaa.mx](mailto:ediazd@correo.uaa.mx), [libarram@gmail.com](mailto:libarram@gmail.com)

Currently, this is one of the most important problems for manufacturing companies. Such algorithms should be efficient and generate feasible decisions and, at the same time, help optimize production plans.

The first solutions for dynamic task scheduling arose within the Operations Research area (Greeding Algorithms [4], Dynamic Programming [14], Branching and Bound [3], [16]), although heuristic techniques (Simulated Annealing [30], Genetic Algorithms [15], Tabu Search [26], Neural Networks [31], Evolutionary Programming [17]) have been the ones that, together, have provided the best solutions.

This paper deals with the dynamic task scheduling within the manufacturing industry, where several objective functions are considered so as to approach more to their actual modeling. Some objective functions that can be optimized are *flow time* (i.e. the time that an job takes until its completion), *tardiness* (i.e. the time exceeded by a job until its termination date), *makespan* (i.e. the duration of a production plan) or *setup* (i.e. task changes during the execution of a production plan). As a benchmark, we will use a real-world problem posed by Morley [18] which involves a General Motors truck plant where one assembly line produces one truck per minute; said truck is to be allocated to a paint booth. Each truck enters with a previously assigned color, based on a job order, and the painting work takes three minutes. The booth takes one additional minute to change colors, which generates both a bottle neck and a waste of paint.

Morley [18], [19] proposed a simple bidding mechanism in which two agents send requests for arriving trucks in accordance with the length of their current queue and the color required by the last truck on the queue. In the simulation, this simple bidding mechanism proved to be more effective than the centralized scheduler that controlled access to paint booths (with an improvement of 10%), taking into consideration only the objective function of the number of color changes (setup).

Other authors have approached the dynamic task scheduling issue. Among them, Cicirello & Smith [9] proposed a solution by using an algorithm based on social insect behavior, namely Wasp Colony. This metaheuristic was first proposed by Theraulaz et al. [27] [28]; it represents a self-organization model that takes place within a wasp colony. The wasp behavior model describes the nature of interactions between wasps as individuals and their local environment with respect to task assignment. Their colony self-organized task scheduling model uses what is referred to as *response thresholds*. An individual wasp has a response for each wasp nest zone. Based on the wasp threshold for a given zone and the amount of stimulus of the brood located in that zone, a wasp can or cannot undertake the feeding task in such location. A lower threshold in a given zone equals a greater probability of undertaking an activity when a stimulus is given.

The Cicirello & Smith [9] model integrates two algorithms, one for allocating work to resources, namely routing wasp, and another one for scheduling production, namely wasp scheduler. Like Morley [19] Cicirello [9] only looks forward to optimizing the number of task changes accumulated by the manufacturing system.

Nouyan et al. [21] retake the approach used by regarding Cicirello & Smith [9] Morley's model [19] and propose four modifications to said algorithm: the manner in which the thresholds are refreshed, task priority calculation, probability

calculation so as to obtain a job, and a way to foster idle resources to participate in task request. They also extend the dynamic threshold model used by Theraulaz et al. [27] in order to solve the heterogeneous case in which different booths may require different processing times for completing a similar task.

Cao et al. [8] also worked based on the approach made by Cicirello & Smith [9] and modify both the routing algorithm, changing the form in which the probability of selecting a task is calculated, and the scheduling algorithm, modifying how hard the bidding is done for certain task. Also, they propose a multiagent system architecture for implementing their approach.

In Cicirello & Smith [9], the function to be optimized is the minimization of color changes in the booths; in Nouyan et al. [21] and in Cao et al. [8], the function to be optimized is the minimization of the makespan.

The objective of this paper is to develop a hybrid algorithm that combines the Wasp Colony bio-inspired metaheuristics with a set of priority rules so as to attain the simultaneous optimization of the objective functions: setup, tardiness and makespan, using the problem posed by Morley [18] which is located within a dynamic task scheduling environment at a manufacturing company.

Section 2 represents the task scheduling issue, its model and characteristics. Section 3 shows the production system elements, their constraints and resource operation rules. Section 4 shows our solution proposal, based on a hybrid between the routing wasp algorithm and a set of heuristic rules that perform dynamic task scheduling, and defines the applied multiobjective focus. Section 5 shows an experimental analysis that presents improvements in performance obtained by the hybridization proposed with respect to the previous work [9]. Section 6 presents our conclusions.

## 2 Job Shop Scheduling Problem

The job shop scheduling is an optimization issue which can be formalized through its mathematical model [14]:

Given sets:  $\mathcal{M} = \{M_1, M_2, \dots, M_m\}$  of machines,  $\mathcal{J} = \{J_1, \dots, J_n\}$  of jobs, and  $\mathcal{O} = \{O_{ij} | i = 1, \dots, \mu_j, j = 1, \dots, n\}$  operations.

Where  $m$  is the number of machines,  $n$  is the number of tasks,  $\mu_j$  is the number of operations of task  $J_j$ , and  $\mu = \max_j \mu_j$ .

Operation  $O_{ij}$  is the  $i^{\text{th}}$  operation of task  $J_j$  which requires a given processing time over machine  $M_{\kappa_{ij}} \in \mathcal{M}$ , where  $\kappa_{ij}$  indicates the machine that shall process operation  $O_{ij}$ , for an uninterrupted period of a given  $p_{ij}$  length.

Should the time to complete operation  $O_{ij}$  be defined as  $C_{ij}$ , then one of the possible objective functions is to produce a plan that minimizes the maximum termination time,  $C_{\max} = \max_{i,j} C_{ij}$  whose optimum value is defined by  $C_{\max}^*$  also named *makespan*.

Operation  $O_{ij}$  requires the exclusive use of  $M_j$  in an uninterrupted duration  $p_{ij}$ , which is the *processing time*, and where *preemption* is not allowed.



The problem is characterized as follows: Each *machine* can only process one operation at a time, and each task can only be performed by not more than one *machine* at a time in an order provided by a predefined *technological sequence* of machines.

The *starting* and *due times* of one operation  $O_{ij}$  is denoted by  $s_{ij}$  and  $c_{ij}$  respectively, and where a *schedule* is a set of *due times* of each operation  $\{c_{ij} | 1 \leq i \leq n, 1 \leq j \leq m\}$  that satisfies their constraints.

The predefined *technological sequence* for each *job* can be collectively given as a matrix  $\{T_{jk}\}$  in which  $T_{jk} = r$  corresponds to the  $k^{\text{th}}$  operation  $O_{jr}$  of *job*  $J_i$  of *machine*  $M_r$ .

This is a rather wide problem that may derive in different subproblems depending on the considered restrictions and characteristics of each case, reason why most of them are complete NP problems [2].

The description of a specific task planning problem in a plant can be performed through the following information:

- The description of the job arrival process (*A*)
- The number and type of machines that integrate the job shop (*B*)
- The description of the plant's floor flow pattern (*C*)
- The criteria with which these production plans are assessed (*D*)

The notation used by these for parameters so as to define a planning problem is then  $A/B/C/D$  [10].

For Morley's [18], problem, including our solution proposal, the notation is:

$$n/m/R/C_{max}, T_{min}, S_{min}$$

where  $n$  is the number of trucks to be painted,  $m$  is the number of paint booths,  $R$  is the random arrival of trucks to the booths and, last, the objective functions to be optimized: makespan, tardiness and number of paint color changes.

In this case, the integrated workshop production system plant flow problem is defined by the number of parallel multipurpose machines (paint booths) that can process the same number of different types of works (painting trucks in different colors), although with a refreshing cost by machine, from one type of job to the other.

### 3 Plant Description in Morley's Problem

Morley's problem [18], involves the dynamic allocation of jobs to booths in a real General Motors factory, where the production line produces trucks that enter the paint plant and are allocated to the queue of some of the booths located at the end of the manufacturing line and where each queue has a truck limit. When the queues are full, the following truck should wait in a storage area until it can be processed by the system.

The production system that will be worked on is formed by the different elements:

- One assembly line where trucks ready to be painted come from
- One series of paint booths capable of painting in different colors
- One area where trucks that cannot be serviced by the booths are stored in

**Constraints in Morley's Production System:** The production system allocates each truck that leaves the assembly line to one paint booth, taking into consideration the following elements:

- The time of arrival to a paint plant is of one truck per time unit.
- There are seven booths in the paint booth.
- Each booth has a process queue for 3 trucks at most.
- Painting a truck takes 3 time units.
- There are 14 different paint colors.
- Each truck requires a specific paint color and arrives with no order.
- Approximately 50% of trucks require the same color.
- The remaining 50% requires colors randomly distributed among the other 13 colors.
- One paint booth can only use one color at a time and there is a paint update cost.
- The update cost is of one time unit.

**Paint Booth Operation Rules:** the following three considered rules are established in this order:

- Allocate one truck to the booth with shorter queue (with space) whose last truck is of the same color as the next one (if such booth exists).
- Allocate the truck to the booth with an empty queue, if there is one.
- Allocate the truck to the booth with a shorter queue (with space), if there is one.

Each booth uses one color at a time and changing a color requires a cost related to the time necessary for making the change (and also to the wasted paint). When a color update is made, the wasted paint amounts are significant (between 1/6 and 1/8 of that required for each vehicle, according to Braslaw [7]). This shows the importance of handling a control policy when allocating trucks to paint booths which minimizes color changes. Attaining this objective is not easy, as the colors required by the next trucks are not known and there are unexpected events, such as booth failures and others, present.

## 4 Solution Proposal

Our solution proposal is based on a multiagent system which is formed by agents related to the booths that bid for the trucks that arrive from the assembly line, and where the winning agent is the one presenting a greater probability value. When all agents are busy, the trucks are sent to a warehouse (storage), where they are

allocated to virtual queues in accordance with their color; every certain time, booths that paint with the same color are searched for so as to be assigned warehouse truck groups with that same color. For the job routing algorithm, we start from the solution approach proposed by Theraulaz [27] [6], which is based on the Wasp Colony Algorithm and, for the scheduling function, we added a series of priority rules that make up a local optimizer, which controls the allocation of trucks that are in the warehouse, taking into consideration a multiobjective focus for improving the system's performance

#### **4.1 Wasp Colony Algorithm**

Theraulaz' [27] work shows a self-organization model inside a wasp colony, where the author describes the behavior and nature of the interactions between wasps, as individuals, and the local environment with respect to the allocation of tasks through response thresholds. Each wasp has a response threshold for each nest zone and, depending on the amount of stimulus that is emitted by the brood located in that zone; the wasp may or may not perform feeding tasks. A low threshold in a given zone amounts to a greater probability of undertaking an activity when a stimulus is given. The threshold may be fixed in time [5], or it is possible to consider that the threshold falls during time periods in which the task is performed or increased [6].

In Cicirello & Smith [9], the response threshold is a bias to respond to a stimulus more than a specific response degree. Taking this into consideration, the system uses the wasp's behavior model and the response threshold so as to formulate an adaptive policy so as to decide whether to bid or not for a job.

In the wasp behavior model, when two individuals of a colony meet, there is a possibility that they interact in a dominance competition. Should this interaction occur, the wasp with higher social rank has greater probability of dominating the interaction. Through these interactions, the colony's wasps self-organize in a dominance hierarchy.

[27] incorporates this aspect to its system in order to determine the winning bid. In other words, when two or more routing wasp choose the same job, the winner is selected through the dominance bidding. The wasp colony model puts two algorithms together: one routing wasp (which updates the thresholds and decides what job it will bid for) and a wasp Scheduler (which determines the priority of each job).

#### **4.2 Routing Wasp Algorithm**

In the problem posed by Cicirello & Smith [9], each booth has a routing wasp related to it; said routing selects the jobs that are to be bid for so as to attain a possible allocation to its queue, and each routing wasp has a set of thresholds similar to those in the actual wasp model.

#### 4.2.1 The Routing Wasp Behavior Mathematical Model:

The routing is in charge of selecting which truck it will request for a possible allocation to its related booth queue. Each routing wasp has a set of thresholds very similar to the underlying behavior model of an actual wasp:

$$\Theta_w = \{\theta_{w,0}, \dots, \theta_{w,j}\} \quad (1)$$

where  $\theta_{w,j}$  is the response threshold of routing wasp  $w$  to jobs of type  $j$ .

The trucks inside the system that have not been allocated to a booth and that await to be allocated irradiate a stimulus  $S_j$  to all the routing wasp; said stimulus is proportional to the time awaited by the truck for an allocation to a booth. The longer the truck stays without an allocation, the stronger the stimulus it will emit. A routing wasp  $w$  may bid for a job that emits a stimulus  $S_j$  with probability:

$$P(\text{bid}|\theta_{w,j}, S_j) = \frac{S_j^2}{S_j^2 + \theta_{w,j}^2} \quad (2)$$

Otherwise, it should not bid. Threshold values  $\theta_{w,j}$  may vary within range  $[\theta_{min}, \theta_{max}]$ . At all times, each routing wasp knows what its booth is doing, including: the color of the truck that is being processed, the state of its queue, whether the booth is performing an update or not, and whether the booth is idle or not. This response threshold update occurs in every step. Should the booth be currently processing a truck of color  $j$  or should it be in the process of updating to a truck of color  $j$ , then  $\theta_{w,j}$  is updated in accordance with:

$$\theta_{w,j} = \{\theta_{w,0} - \delta_1\} \quad (3)$$

Should the booth be processing or updating painting with a color different than  $i$  then the update is performed in accordance with:

$$\theta_{w,j} = \{\theta_{w,0} + \delta_2\} \quad (4)$$

And, should the booth be currently idle and its queue be empty then, for all the trucks of color  $i$ , the routing wasp adjusts response threshold  $\theta_{w,j}$  in accordance with a:

$$\theta_{w,j} = \{\theta_{w,0} - \delta_3^t\} \quad (5)$$

where  $t$  is the time the booth has been idle and is an exponent, and  $\delta_1$ ,  $\delta_2$  and  $\delta_3$  are system constants. This is why the response thresholds currently in progress for the color truck are reinforced for encouraging the routing wasp to make a bid for a similar color; in the meanwhile, the response thresholds for trucks of other colors that are currently not being worked are adapted so as to discourage routing wasp from bidding for those other colors. This routing wasp specialization (i.e., that of the booths) helps to minimize the update times. The first two forms in which the response threshold is updated (equations 3 and 4) are analog to those of the actual wasp model ([5], [28]). The third equation (equation 5) is included for encouraging

a routing wasp related to an idle machine to engage any job that it may have in order not to remain idle. This last update rule acknowledges that, although specialization may reduce the update time, overspecialization of a type of job with low demand may result in a low system response.

### 4.3 Priority Rules for Scheduling

Once the form of allocation of trucks through the routing wasp algorithm to booths is defined, the following priority rules are established for scheduling the jobs:

- (i) If the truck was not routed to any booth, it is allocated to the waiting queue (warehouse) of the system and placed within a logic queue depending on its color
- (ii) Its stimulus  $S$  is increased by steps as the waiting time passes along.
- (iii) Once 4 trucks of the same color have been accumulated in the waiting queue, a revision is made whether any of the booths is working with that color and the trucks are allocated to said queue with a stimulus value greater than those that may be at the process queue of said booth, with which they get before any others that may be on that same queue and enter directly into the booth.

### 4.4 The Multiobjective Optimization Focus

Generally, the literature on this subject (production scheduling) looks forward to optimizing only one objective function. For instance, scope, delay percentage, etc. This is due to the difficulty of the problem when more objective functions to be optimized are added at the same time, although, in many cases, there are real problems that involve two or more conflictive objectives.

The multiobjective optimization problem is such that it is necessary to find the vector of decision variables that optimize one objective function vector and satisfies some restrictions.

According to [1], it is necessary to optimize vector  $F(\vec{x})$  subject to:

$$\Omega = \{ \vec{x} \in R^n | G(\vec{x}) \leq 0 \} \quad (6)$$

Where:

$\vec{x}$  is the vector of decision variables  $(x_1, \dots, x_n)$

$F(\vec{x})$  is one vector of objective functions  $(f_1(\vec{x}), \dots, f_K(\vec{x}))$

which are functions over  $R^n$  and  $\Omega$  is a non-empty set in  $R^n$

Vector  $G(\vec{x})$  represents the restrictions that may be explicitly handled.

As for the decision-making form, the multiobjective optimization problem can be faced in the following ways [29]:

- *A priori* (Decide  $\rightarrow$  Search) Decision making precedes optimization, combines the different objectives in one step cost function, and converts the problem into a single-objective issue.

- *A posteriori* (Search  $\rightarrow$  Decide) Decision making is made after the optimization has reached a set of equally feasible solutions, named Pareto's optimal set.
- *Progressive* (Search  $\leftrightarrow$  Decide) Decision making is performed together with optimization, through the supply of partial information over the precedence of criteria. This information is integrated within the same solution algorithm. It is an intermediate focus between the other two.

In this case, we employed the *a posteriori* focus, generating a Pareto front of solutions and then that which improves the number of updates (minimum), the average tardiness of jobs (minimum) and the production output (maximum) is selected.

## 4.5 System Modeling

Once the proposed solution components were defined, we created a model so as to implement the computer system that simulates the manufacturing process from Morley's [19] problem. In order to create the model, we will use modeling tool AUML [13] which is an extension of UML which allows the developer to pass from software development to agent development. There are examples of the model components shown below.

**Cases of use:** Use cases are means for specifying the system's required uses [12]; in the following case, paint booths and the wasp router are related through the process of bidding for one truck that leaves the assembly line.

**Process name:** Choose truck to bid

**Description:** Each booth in the system has an associated Routing wasp.

Each Routing wasp is in charge of selecting which trucks it will bid associated booth.

**Actors:** Assembly line, Routing wasp, Paint booth.

**Trigger:** After bidding for a truck, the Routing wasp allocates it to its booth.

**Normal course:** 1. The assembly line sends a truck to the queues of booths for its paintwork  
2. Routing Wasp bid for jobs.  
3. The winning Routing Wasp allocates the truck to the booth of its associated booth.

**Alternate course:** None

The following example of case of use shows the activity that involves sending the truck to the warehouse between the assembly line and routing wasp.

**Process name:** Send truck to storage

**Description:** If the Wasp Router does not accept the truck, then it is sent to the warehouse.

Each wasp router is in charge of selecting which trucks it will bid for; nevertheless, if none of them has space in its queue, the truck is sent to a warehouse.

**Actors:** Assembly line, Routing wasp, Warehouse.

**Trigger:** If no router accepts the truck, it is then sent to the production system warehouse.

**Normal course:** 1. The assembly line sends a truck to the queues of booths for its paintwork.

2. Routing wasp do not bid for the truck.

3. The truck is sent to the warehouse.

**Alternate course:** None

The following example of case of use shows the activity that involves sending a truck of color  $i$  from the assembly line to the routing wasp.

**Process name:** Send truck to color  $i$

**Description:** From the assembly line, each truck may require any of the 14 colors and arrive without any order in particular

**Actors:** Assembly line, Routing wasp

**Trigger:** The arrival of a truck with a specific color.

**Normal Course:** 1. A truck above the assembly line.

2. The assembly line sends a truck of color  $i$  to all the routings wasp.

**Alternate Course:** None

**Class diagrams.** Taking into consideration the previous characteristics, the agent class diagram [12] where agent stereotypes were employed (in this case, reactive agents that respond to environmental changes) was prepared, specifying the following attributes:

- Perception - through which the agent receives physical or social information from the environment (private).
- Interaction - enables the agent to interact and communicate with others in direct or indirect (public) form.
- Abilities - specific knowledge of the agent for performing actions (private).
- Action - actions performed over the environment (private).

The assembly line, paint booths and warehouse are objects that compose the environment of the multiagent system, which has association relationships, although said relationship is no longer strong, reason why its life lines do not depend on one another. Figure 1 shows an Agent Class diagram

**State Diagram.** The state diagram provides the graphical representation of discrete behavior through finite state transition systems and is focused on the changing state of the system that is controlled by events. In this case, the example shows the states that may be assumed by each paint booth, which go from idle up to full queue.

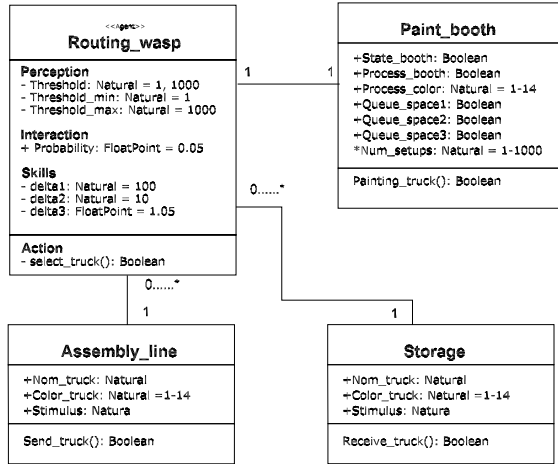


Fig. 1 Agent Class Diagram

**Interaction Diagram.** The interaction diagram shows the interaction focused over the sequence of messages that are exchanged, along the occurrences of their corresponding events over the life lines [23]. Figure 2 shows the diagram of interaction between the Assembly Line, Routing Wasp, Paint Booth and Warehouse (Storage).

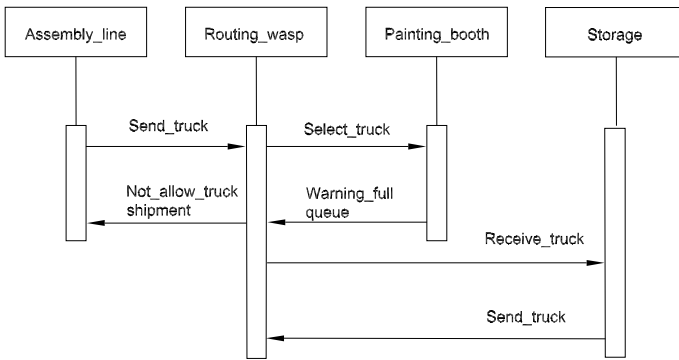
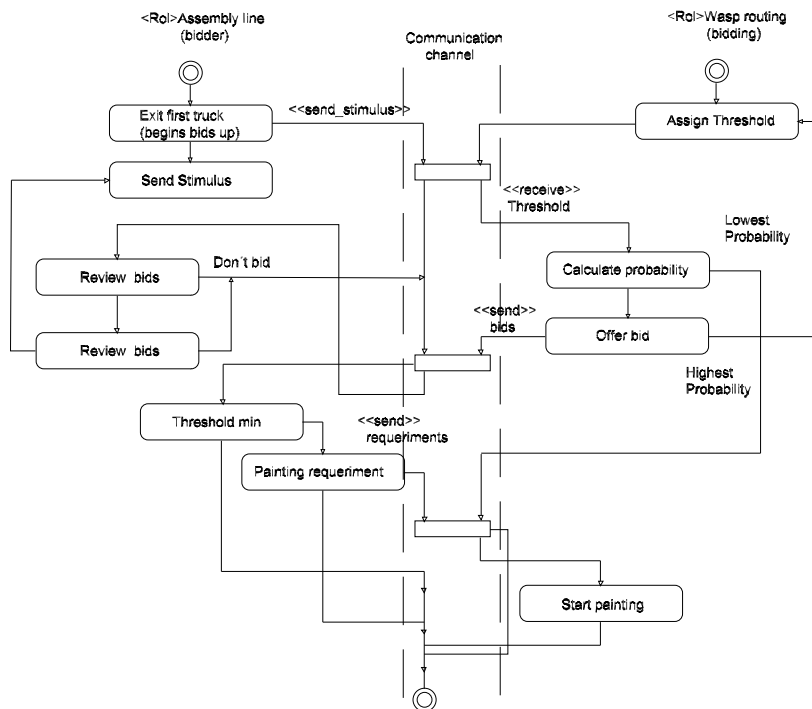


Fig. 2 Interaction diagram

**Activity Diagram.** The protocol between elements will be specified through an activity diagram [22] the truck allocation process uses a paradigm based on the market where the wasp router bids for taking trucks of the assembly line. This triggers a stimulus mechanism in time as it passes along; if this is not taken into account, the stimulus increases. The generated protocol is shown in figure 3.





**Fig. 3** Routing Bidding Process Activity Diagram

**Time Domain Modeling.** In this part, the times and rules of propagation as time passes along are described for the model. Temporary restrictions are related to the time trucks have to stay in the system [11]. Here, we define the type of temporary restrictions considered in this work.

- Time Domain: *Discrete*
- Propagation Rule: *Periodical*
- Dependences between Time Domain: *Synchronized*
- Types of Temporary Constraints: *Timed*

## 5 Experimental Design

A description of the implemented problem is shown below.

Problem parameters:

Seven paint booths and 14 colors.

Each booth starts with a randomly selected color. The queue is limited to three trucks.

Trucks arrive, one by time unit.

The painting time is of three time units per truck.

If the queue is full, the Routing Wasp will try to prevent its queue from reacting to the stimuli from the trucks.

Initial values  $\theta_{w,j}$  are randomly selected in relation to initial booth  $w$  and the painting color.  $c$ .

If  $c = j$  then the initial value for  $\theta_{w,j}$  is  $\theta_{min}$

In any other case, it is randomly selected from the interval  $[\frac{\theta_{max}}{2}, \theta_{max}]$

If a truck  $j$  cannot be assigned to any of the booths  $k$ , it is sent to the waiting queue (warehouse) and it is assigned to the virtual queue of its same color.

The stimulus  $S$  that is emitted by a truck  $j$  is increased by steps as time units  $t_x$  pass along in accordance with the following pattern:

$$S_j = (1\{t_1, t_2\}, 2\{t_3, t_4\}, 3\{t_5, t_6\}, 4\{t_7, t_8\}, 5\{t_9, t_{10}\}, 6\{t_{11}, t_{12}\}, 7\{t_{13}, t_{14}\})$$

The algorithm parameters are defined as:

The update of thresholds occurs during 1/4 of the time unit. The simulation experiment was run during 1000 time units

- $\delta_1 = 100$
- $\delta_2 = 10$
- $\delta_3 = 1.05$

**Table 1** System Performance

> Performance measures >	Cicirello & Smith Algorithms	> Multi-objective
Setups	302	171
Average tardiness	6	6.9
Makespan	994	967

The result was obtained through 100 independent simulation runs and table 1 shows the average values of the updates (paint changes), tardiness, production times and total throughputs.

## 6 Conclusions

This paper presents a hybrid job shop scheduling system originally based on the works of Bonabeu [6] and Cicirello & Smith [9] applied to a real problem (Morley [18]).

The solution is obtained by an algorithm referred to as wasp colony, based in the natural behavior of these insect societies. The algorithm is integrated by a routing and a scheduler; this work employs the routing specified in Cicirello & Smith [9] and the scheduler's algorithm is modified through a series of priority rules that make

up a local optimizer, which allow decreasing the number of updates required for finishing a lot of products. Also, the modification to used not affect significantly the tardiness of jobs and the total number of finished jobs, looking for a commitment between the objectives.

An analysis of the production system modeling of Morley's problem is realized and the proposed model is implemented and used. The AUML language is used in order to make up the multiagent system that operates in the job shop scheduling problem.

As a future work, we intend to incorporate the analysis of cases in which resources may fail and verify the robustness of the proposal.

## References

1. Abbas, H.A.: The self-adaptive Pareto differential evolution algorithm (online). In: Congress on Evolutionary Computation, Piscataway, NJ, USA, vol. 1, pp. 831–836 (2000)
2. Applegate, D., Cook, W.: A Computational Study of the Job Shop Scheduling problem. *ORSA Journal on Computing* 3(2), 149–156 (1991)
3. Barker, J.R., McMahon, G.B.: Scheduling the General Job-Shop. *Management Science* 31(5), 594–598 (1985)
4. Binato, S., Hery, W., Loewenstern, D.Y., Resende, M.: A GRASP for Job Scheduling. Technical Report N 00.6.1 AT&T Labs Research (1999–200)
5. Bonabeau, E., Theraulaz, G., Deneubourg, J.L.: Fixed response threshold and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology* 60, 753–807 (1998)
6. Bonabeau, E., Sobkowski, A., Theraulaz, G., Deneubourg, J.L.: Adaptive task allocation inspired by a model of division of labor in social insects. In: Lundh, D., Olsson, B. (eds.) *Bio Computation and Emergent Computing*, pp. 36–45. World Scientific (1997)
7. Braslaw, J.: Personal communication, Material Sciences Department, Ford Research (2001)
8. Cao, Y., Yang, Y., Wang, H., Yang, L.: Intelligent Job Shop Scheduling Based on MAS and Integrated Routing Wasp Algorithm and Scheduling Wasp Algorithm. *Journal of Software* 4(5) (2009)
9. Cicirello, V.A., Smith, S.F.: Wasp-like Agents for Distributed Factory Coordination. The Robotics Institute CMU (2001)
10. Conway, R.W., Maxwell, W.L., Miller, L.W.: *Theory of Scheduling*, pp. 6–8. Addison Wesley (1995)
11. FIPA, Foundation for Intelligent Physical Agents: FIPA Modeling Area: Temporal Constraints (2003)
12. FIPA, Foundation for Intelligent Physical Agents: FIPA Modeling Area: Interaction Diagrams (2003)
13. Gutierrez Marfileno Luis, F., de Leon Eunice, P.: Modelado de un sistema en tiempo real mediante AUML, 11o Seminario de Investigacion. Universidad Autonoma de Aguascalientes (2010)
14. Hillier, F.S., Lieberman, G.J.: *Introduction to Operations Research*, 8th edn., pp. 440–466. McGraw Hill (2006)

15. Omar, M., Baharum, A., Hasan, Y.A.: A job-shop scheduling problem (JSSP) using Genetic Algorithm (GA). In: Proceedings of 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications University Sains Malaysia, Penang (June 2006)
16. Manne, A.S.: On the Job-Shop Scheduling Problem. *Operations Research* 8, 219–223 (1960)
17. Hammadi, M.K., Slim, B.P.: Evolutionary Algorithms For Job-Shop Scheduling. *Int. J. Appl. Math. Comput. Sci.* 14(1), 91–103 (2004)
18. Morley, D.: Painting trucks at general motors: The effectiveness of a complexity-based approach. *Embracing Complexity: Exploring the Application of Complex Adaptive Systems to Business*, The Ernst and Young Center for Business Innovation, pp. 53–58 (1996)
19. Morley, D., Schelberg, C.: An analysis of a plant-specific dynamic scheduler. In: Proceedings of the NSF Workshop on Dynamic Scheduling, pp. 115–122 (1993)
20. Nabil, N., Elsayed, E.A.: Job shop scheduling with alternative machines. *International Journal of Production Research* 28(9), 1595–1609 (1990)
21. Nouyan, S.: Agent-Based Approach to Dynamic Task Allocation. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) ANTS 2002. LNCS, vol. 2463, pp. 28–39. Springer, Heidelberg (2002)
22. Odell, J., Parunak, V.D., Bauer, B.: Extending UML for Agents. In: AOIS Workshop on AAAI 2000 (2000)
23. Picard, G.: UML Stereotypes Definition and AUML Notations for ADELFE Methodology with OpenTool. In: First European Workshop on Multi-Agent Systems. St. Catherine College Oxford (2003)
24. Pinedo, M.: *Scheduling: Theory, Algorithms and Systems*, 3rd edn., p. 1. Springer (2008)
25. Stancovic, A.: Misconceptions about Real Time Computing. *IEE Computer*, 10–19 (1988)
26. Taillard, E.: Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. *ORSA Journal on Computing* 6, 108–117 (1994)
27. Theraulaz, G., Goss, S., Gervet, J., Deneubourg, J.L.: Task differentiation in polistes wasp colonies. In: Proc. of the First Intl. Conf. on Simulation of Adaptive Behavior. MIT Press (1991)
28. Theraulaz, G., Bonabeau, E., Deneubourg, J.L.: Response threshold reinforcement and division of labour in insect societies. *Proc. R. Soc. London B* 265(1393), 327–335 (1998)
29. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evolutionary Computation* 8(2), 125–147 (2000)
30. Takeshi, Y., Ryohei, N.: *Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search*, pp. 237–248. Kluwer academic Publishers, MA (1996)
31. Zhou, D.N., Cherkassky, V., Baldwin, T.R., Olson, D.E.: A Neural Network Approach to Job-shop Scheduling. *IEEE Trans. Neural Networks* 2(1), 175–179 (1991)

# Optimization Metaheuristic for Software Testing

Nashat Mansour, Hratch Zeitunlian, and Abbas Tarhini

**Abstract.** This paper presents an evolutionary method for testing web applications. Although state-based testing has been reported, few papers have addressed modern web applications. In our work, we model web applications by associating features or web pages with states; state transition diagrams are based on events representing state transitions. We formulate the web application testing problem as an optimization problem and use a simulated annealing (SA) metaheuristic algorithm to generate test cases as sequences of events while keeping the test suite size reasonable. SA evolves a solution by minimizing a function that is based on the contradictory objectives of coverage of events, diversity of events covered, and definite continuity of events. Our experimental results show that the proposed simultaneous-operation SA gives better results than an incremental SA version and significantly better than a greedy algorithm.

## 1 Introduction

During the past decade web applications evolved significantly. A new dimension of web technology, known as Web 2.0, was depicted where web applications are no longer static pages but lighter client applications. In Web 2.0 the web is approached as a platform, and software applications are built upon the web as opposed to be built upon the desktop [1]. The web development technologies changed into technologies that rely heavily on client side code.

Web 2.0 applications are heavily built around several technologies such as AJAX (Asynchronous JavaScript and XML), rich media content, widgets, and third party applications that can be executed within webpages, Webparts, Portlets and similar HTML units. Applications developed with AJAX technology provide the user with a

---

Nashat Mansour · Hratch Zeitunlian · Abbas Tarhini

Department of Computer Science and Mathematics,

Lebanese American University, Beirut, Lebanon

e-mail: [nmansour, hratch.zeitunlian, abbas.tarhini}@lau.edu.lb](mailto:{nmansour, hratch.zeitunlian, abbas.tarhini}@lau.edu.lb)

rich dynamic interface that enables responsive interaction through light client software where the user is capable of controlling the content of the website through asynchronous request and responses resulting in a new page that is updated dynamically through the Document Object Model (DOM) [2].

The new Web technology introduces additional challenges to the already hard task of Web application testing. In addition to the searchability and accessibility, we have to test the dynamic user interface elements and states to find abnormalities and errors [3]. Thus, existing Web testing methods [4,5,6] are not adequate to Test Web 2.0 applications.

In this paper, we propose an effective state-based testing method which is designed to handle the complexity of the Web 2.0 application. This method is based on deducing Web page states and generating the equivalent state chart. Then, we use a metaheuristic approach based on simulated annealing to simultaneously generate a controlled number of test cases with maximum diversity and coverage, and then compare the results with those of other strategies. Simulated annealing is a single-solution-based well-established metaheuristic that has been used for solving many real-world problems. It has exhibited faster processing than population-based metaheuristics [7,8].

## 2 Related Work on Testing Web Applications

There exist several functional testing tools proposed by the industry for testing Web application. Some tools rely on capture/replay facilities which allow functional testing [9]. They record the interactions that users have with the graphical interface and repeat them during regression testing. However, they do not detect the failures in meeting the functional requirements. Other tools rely on discovering and systematically exploring Web-site execution paths that can be followed by a user in a Web application [10]. Further approaches to functional testing are based on user session data to produce test suites [11] others are based on HttpUnits where the application is divided to HttpUnits and tested mimicking web browser behavior [12]. HttpUnit can be used for unit testing and it is best suited for the implementation of functional tests and acceptance tests; however, it is not practical for typical Web layer components like JSP pages, servlets, and other template components. Liu et al. [13] propose a formal technique that models web application components as objects and generates test cases based on data flow between these objects. Ricca and Tonella [14] present a test generation model based on the Unified Modeling Language. In fact, these techniques extend traditional path-based test generation and use forms of model-based testing, but can be classified as “white-box” since the testing models are generated from the web application code.

The major challenges for the techniques of testing web applications with dynamic features are how to model the application and what algorithm can be used in order to select the test cases from a huge number of possibilities. Not much research has been reported on testing web applications with dynamic features using state transition diagrams. Marchetto et al. [15] proposed a state-based testing technique

designed to address the new features of Web 2.0 applications. In this technique, the DOM manipulated by Ajax code is abstracted into a state model where callback executions triggered by asynchronous messages received from the Web server are associated with state transitions. The test cases are generated from the state model based on the notion of semantically interacting events. Empirical evidence shows the effectiveness of this kind of testing in finding faults. However, this technique generates a very large number of test cases that could limit the usefulness of the test suites. Another proposal by Marchetto et al. [3] addressed this problem. They proposed a search-based approach based on a hill-climbing algorithm to generate test sequences while keeping the test suite size reasonably small. In order to preserve a fault revealing power comparable to that of exhaustive test suite, they aimed to maximize the diversity of the test cases by introducing a measure of test case diversity instead of exhaustively generating all test cases up to a given length  $k$ , and selecting the most diverse test cases, without any constraint on their length  $k$ .

### 3 Testing Web Applications

The dynamic features of Web 2.0 add more complexity to the already hard task of web application testing. The complexity is found in the absence of traditional navigation paths. A complete Web 2.0 web application can be made from a single page whose content and functionalities change by asynchronous server calls raised by the user interaction with the application, which changes the state in the client site, resulting in a dynamic DOM. It is not possible to walk through the different states of the dynamic page since there is no unique URI assigned to a specific variant of the dynamic page, unlike in traditional web applications where we have an explicit and unique URI for each Web page and each variant of a dynamic web page.

To test Web 2.0 applications and to cover the dynamic aspects of the Web 2.0, widgets, third party applications that can be executed within WebPages, Webparts, Portlets and hypermedia, we suggest a state based testing strategy that will dynamically generate a finite state machine from a web application by extracting semantically interacting events [15] that produce state changed in the user interface. From the inferred graph test cases, a sequence of events will be generated. However, generating test case sequences from the finite state machine can lead to a very large number of test cases in the test suites. This is why Marchetto et al. [3,15] suggested a search based approach to generate long sequences of events while keeping the test suite size reasonably large using a hill-climbing algorithm. The problem with this algorithm is that the solution will be a local optimum rather than being a global optimum.

The objective of our research is to develop a more effective state based testing for a Web 2.0 application that will cover its dynamic features. This testing approach is based on a search-based algorithm rather than exact graph algorithms for traversing the events in the state-based graph model. The reasons behind this decision are: (a) often, when browsing or traversing through the functionalities of a web application, we end up visiting the same application state more than once, thus resulting in a

partial loop-like sequence; (b) the objective of our research is develop optimal or good suboptimal test suite that reduces the number of test cases and not merely a set of sequences of events/edges in the graph; (c) although depth-first-search (DFS) graph algorithms can guarantee graph coverage but they do not guarantee diversity, nor do they allow elimination of some events/edges; (d) DFS would generate a number of test cases but without prioritization and without including significance weights of events; (e) DFS will start from a root, i.e. test sequences will start from the same state, which is not the case in web applications where users can start traversing the site from any state through a saved hyperlink; (f) DFS does not allow loop traversals.

## 4 State Graph Modeling

Extracting a state graph form a Web 2.0 application is not a direct and simple task. The main challenge is the absence of traditional navigational paths. This is because in Web 2.0 there is no unique URI assignment to a specific variant of the Dynamic Page, unlike traditional web applications where each web page state in the browser has an explicit URI assigned to it. Moreover, an entire Web 2.0 application can be created from a single web page where User Interface (UI) is determined dynamically through changes in the DOM initiated by user interaction through asynchronous server calls. Further, Web 2.0 application may contain third party HTML units, user shared data, widgets and media content that are added to the application simultaneously during execution. To overcome the above mentioned challenges our testing mechanism will reconstruct the user interface states, and generate static pages having navigation paths each with unique URL. These static pages will be used to conduct state-based testing [15].

To achieve the static-like pages we need a tool that will execute client side code, and identify clickable elements which may change the state HTML/ DOM within the browser. From these state changes, we will build our state graph that captures the states of the user interface, and the possible transitions between the states.

### 4.1 State Graph

Our model must reveal all user interface state changes in Web 2.0 application. Thus the model must record all navigation paths/semantically interacting event of the DOM state changes. This can best be represented by a State Graph which is defined as follows:

**Definition 1** A State Graph for a Web 2.0 site  $A$  is a 4-tuple  $\langle r, V, C, E \rangle$  where:

- (i)  $r$  is the root node representing the initial state after  $A$  has been fully loaded into the browser.
- (ii)  $V$  is a set of vertices representing the states. Each  $v \in V$  represents a run-time state in  $A$ .



- (iii)  $C$  is a set of clickable elements that enables the transition from one state to another.
- (iv)  $E$  is a set of edges between vertices. Each  $(v_1, v_2) \in E$  represents a clickable  $c \in C$  connecting two states if and only if state  $v_2$  is reached by executing  $c$  in state  $v_1$ .

**Definition 0.1.** Semantically interacting events: Events  $e_1$  and  $e_2$  are interacting semantically if there exists as state  $S_0$  such that their execution in  $S_0$  does not commute, i.e., the following conditions hold:

$$\begin{aligned} S_0 & \Rightarrow_{e_1:e_2} S_1 \\ S_0 & \Rightarrow_{e_2:e_1} S_2 \\ S_1 & \not\leftrightarrow S_2 \end{aligned}$$

where  $S_0$ ,  $S_1$ , and  $S_2$  are any states in the state graph of the web application.

The notion of pair of semantically interacting events can be easily generalized to sequences.

**Definition 0.2.** Sequence of semantically interacting events: The event sequences  $(e_1, \dots, e_n)$  is a sequence of semantically interacting event if every pair of events in the sequence is pair of semantically interacting events according to [Definition 0.1](#).

## 4.2 Building the State Graph

Two issues are to be considered while building the state graph. First we need to detect the event-driven elements; next, we need to identify the state changes. The state graph is created incrementally; initially, the state graph contains only the root state. Additional states are appended to the graph as event-driven elements are traced/invoked in the application and state changes are analyzed. In the following subsections, we detail how event-driven elements are detected and state changes are identified.

### 4.2.1 Detecting Event-Driven Elements

In order to detect even-driven elements, we suggest a candidate list of elements that responds to events (clickable events). For example, `<div>`, `<input>`, `<a>`, and others may respond to events like (Click, Doubleclick, Mouseover). Once an HTML page is loaded, we access the HTML elements through the DOM and detect the event-driven elements by checking whether each element in the DOM belongs to the suggested candidate list of clickable events.

### 4.2.2 Identifying State Changes and Inferring the FSM

Once a candidate element is detected, we execute the event attached to that element. In order to determine whether the execution of the event results in state change, we compare the version of DOM-tree after firing the event and the DOM-tree version

just before firing that event. If the execution of the event results in state change, we check whether the resulting state is already covered in the graph. If the state hasn't been covered, the new state is added to the graph, and an edge representing the executed event will connect the two states. If the state is already covered, an edge will be added between the states. Similarly all clickable elements will be detected and executed, and the FSM generated covering all functionalities the Web 2.0 Application included.

## 5 Simulated Annealing Algorithm

The simulated annealing algorithm (SA) simulates the natural phenomenon by a search (perturbations) process in the solution space optimizing some cost function (energy) [16]. It starts with some initial solution at a high (artificial) temperature and then reduces the temperature gradually to a freezing point. In the following subsections, we describe how we generate test sequences of semantically interacting events using the simulated annealing algorithm; an outline of the SA algorithm is given in Algorithms 1 and 2. In our work, we choose simulated annealing, in contrast with hill-climbing [3], in order to generate test sequences because it allows uphill moves which will force the solution to jump out of a local minimum a more promising downhill, in a controlled way.

---

### Algorithm 1 Outline of the SA Algorithm

---

```

1: Initial configuration = Sequence of events from the state graph;
2: Determine initial temperature  $T(0)$ ;
3: Determine freezing temperature  $T_f$ ;
4: while  $T(i) > T_f$  and not converged do
5:   repeat
6:     Generate_function();
7:   until several times (multiple of the number and size of required test cases)
8:   save_best_sofar();
9:    $T(i) = \Theta \cdot T(i)$ ;
10: end while

```

---



---

### Algorithm 2 Procedure Generate\_function()

---

```

1: perturb();
2: if  $\Delta OF1 \leq 0$  then
3:   update(); {/*accept*/}
4: else if random() <  $\exp(-\Delta OF1/T(i))$  then
5:   update(); {/*accept*/}
6: else
7:   reject_perturbation();
8: end if

```

---

## 5.1 Solution Representation

Our solution is represented as a configuration  $C$ , which is implemented as an array of variable-length test cases. Each test case is represented by a sequence of a maximum of  $K$  events derived from the state graph. The length of the array is  $K \cdot N$ , where  $N$  is the maximum number of test cases required in the solution. To allow variable length of test cases, we introduce a random number of fake edges into our set of valid events. These fake edges, called “No Edge”, will play the role of space holder in the array.

## 5.2 Energy Function

The energy function measures how good the current configuration is. We based the energy function on three major weighted factors. The weights represent the importance of each factor. The three factors are Continuity, Diversity and Coverage.

**Continuity:** When testing event based applications it is very important to test a continuous set of events. In fact, test cases with longer continuous sequences of events have higher capability of revealing faults. In our SA strategy, we want to minimize/eliminate the discontinuity ( $DC$ ) of events in a test case. We calculate discontinuity by checking the events in every test case and incrementing the value by one whenever discontinuous events are found.

**Diversity:** Diversity is an important factor which guarantees that test cases will cover events from the entire scope of the web application, and not just concentrate on events from a certain part. Hence, we guarantee equally distributed events within the entire test suite. In this work, we will be minimizing the Lack of Diversity by calculating the average frequency of events in the entire test suite. Thus, given a test suite  $S$ , composed of a set of test cases based on semantically interacting sequences of events, its Lack of diversity ( $LDiv$ ) is computed as follows:

$$LDiv = \sqrt{\sum_{e \in Events} (F_e - F_{avg})^2} \quad (1)$$

where:  $e$  is an event that belongs to the set of events  $Events$ ,  $F_e$  is the execution frequency of event  $e$ , and  $F_{avg}$  is the average frequency of event  $e$  computed over the entire test suite.

**Weighted Coverage:** In Web 2.0 applications, end users and third parties can change the content of a web page dynamically by injecting HTML code or web widgets through their interaction with the site. Thus, some events would have higher importance than other events; accordingly, we may control or even limit some functionality from being included in our testing plan by allowing a measure of importance on events that are part of the original web application, compared to injected

events or functionality into the web application. The importance of events is represented by pre-defined weights assigned to every event. Again we want to minimize the value of the unimportant events and this value is calculated by checking if an event is covered in the test suite and multiplying it with its importance or weight. The weighted coverage is given by

$$WC = \sum_{e \in Events} (W_e \cdot C_e) \quad (2)$$

The Energy function is represented as:

$$E = \alpha \times \frac{1}{WC} + \beta \times LDiv + \gamma \times DC, \quad (3)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are user-defined weights for weighted coverage, diversity, and discontinuity respectively.

Note that different values can be assigned to the weights in  $E$ . These weights are important for selecting test cases. They might be contradictory; that is, by increasing one of these weights, say  $\alpha$ , the solution will improve in minimizing one factor (discontinuity) while it might increase the other factors. These weights will allow flexibility in using our proposed solution algorithms to suit the user's particular choices or requirements for different instances of the problem.

### 5.3 Metropolis Step and Feasibility

An iteration of the Metropolis step, `Generate_function()`, consists of a perturbation operation, an accept/reject criterion, and a thermal equilibrium criterion. Perturbation in our strategy is done randomly by selecting an event within a test case and substituting it with a randomly chosen event from the Events Set.

The acceptance criterion checks the change in  $E$  due to the perturbation. If the change decreases the objective function, the perturbation is accepted and  $C$  is updated. However, if the perturbation causes the objective function to increase, it is accepted only with a probability  $e^{-\Delta OF1/T(i)}$ . The main advantage of this Monte Carlo algorithm is that the controlled uphill moves can prevent the system from being prematurely trapped in a bad local minimum-energy state. Note that for lower temperature values  $T(i)$ , the probability of accepting uphill moves becomes smaller; at very low (near-freezing) temperatures, uphill moves are no longer accepted. The perturbation-acceptance step is repeated many times at every temperature after which thermal equilibrium is considered to be reached.

Perturbations can make  $C$  infeasible if they violate the definition of continuity. But, the formulation of the energy function  $E$  accounts for this infeasibility problem. The last term in  $E$ ,  $DC$ , can be assigned a large weight,  $\gamma$ , so that infeasibility is severely penalized. Thus, infeasible test cases will be prevented at low temperatures.

## 5.4 Cooling Schedule

The cooling schedule is determined by running a heuristic algorithm that deduces the starting and freezing temperatures with respect to the number of Uphill Jumps. The initial temperature  $T(0)$  is the temperature that yields a high initial acceptance probability of 0.93 for uphill moves. The freezing point is the temperature at which such a probability is very small (2-30), making uphill moves impossible and allowing only downhill moves. The cooling schedule used in this work is simple:  $T(i+1) = \theta \cdot T(i)$ , with  $\theta = 0.95$ .

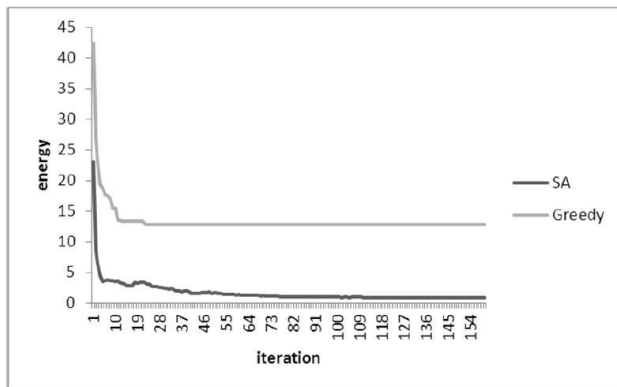
As the annealing algorithm searches the solution space, the best-so-far solution (with the smallest energy value) found is always saved. This guarantees that the output of the algorithm is the best solution it finds regardless of the temperature it terminates at. Convergence is then detected when the algorithm does not improve on the best-so-far solution for a number of temperatures, say 20, in the colder part of the annealing schedule.

## 6 Experimental Results

In this section, we present the results of generating test cases using SA algorithm and compare them with the results of incremental simulated annealing and greedy algorithms. Incremental SA generates test cases one at a time rather than all tests simultaneously. These algorithms are applied on a state graph with 106 events. We generated a test suite of 40 test cases; each test case has a maximum of 6 test events. To simulate variant-length test case size, we appended 10% Fake Edges ('No Edge') to the list of events.

Figure 1 shows the convergence of the energy function for the SA and greedy algorithms. Clearly, SA converges to a lower energy value. Also, the greedy algorithm converges faster (about iteration 20) to a bad local minimum whereas SA takes longer (about iteration 110) to a much better set of test cases.

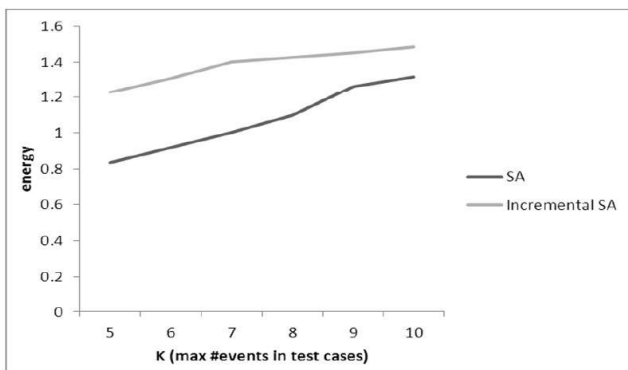
Table 1 shows the comparison between the best energy function for the three strategies (SA, Incremental SA, and Greedy) obtained for different maximum event numbers in a specific test case. The results show that SA converged to the best energy values in all cases followed by incremental SA; the greedy algorithm failed badly in comparison with the two SA versions. Figure 2 graphically compares the results of incremental SA and the (simultaneous-operation) SA algorithm. This figure shows that the incremental SA limits its search by the early decisions on test case selection (made in early iterations), in contrast with the simultaneous-operation SA that simultaneously generates test cases. Also, it shows that as the test case length,  $K$ , increases, the energy values for the three algorithms increase since it becomes harder to maintain relative diversity.



**Fig. 1** The energy function convergence for the SA and greedy algorithms

**Table 1** Energy values for the three algorithms

Max number of events in Test cases	Simulated Annealing	Incremental SA	Greedy
k= 5	0.83314	1.22692	10.36046
k=6	0.91643	1.3067	12.87138
k=7	1.00339	1.40049	16.43683
k=8	1.10088	1.42656	30.36671
k=9	1.26278	1.45067	30.42697
k=10	1.31808	1.48529	36.96823



**Fig. 2** Energy values for the incremental vs simultaneous-operation SA

## 7 Conclusions

We have presented an optimization metaheuristic method for testing web applications. We have also modeled the dynamic features of Web 2.0 using state transition diagrams. We use a simulated annealing algorithm to generate test cases as long sequences of semantically interacting events. We have also formulated an energy function that is based on the capability of test cases to provide high coverage of events, high diversity of events covered, and definite continuity of events. Our experimental results show that the proposed concurrent-operation simulated annealing algorithm gives better results than the incremental simulated annealing and significantly better than a serial greedy algorithm. Future work will aim to compare the SA algorithm with a population-based metaheuristic.

**Acknowledgements.** This work was partially supported by the Lebanese American University.

## References

1. O'Reilly, T.: Design Patterns and Business Models for the Next Generation of Software, <http://oreilly.com/web2/archive/what-is-web-20.html>
2. Document Object Model (DOM), <http://www.w3.org/DOM>
3. Marchetto, A., Tonella, P., Ricca, F.: State-based testing of ajax web applications. In: Proceedings of IEEE International Conference on Software Testing (ICST), Lillehammer, Norway (April 2008)
4. Andrews, A., Offutt, J., Alexander, R.: Testing Web Applications by Modeling with FSMs. *Software and System Modeling* 4(3) (July 2005)
5. Tarhini, A., Mansour, N., Fouchal, H.: Testing and Regression Testing for Web Services Based Applications. *International Journal of Computing & Information Technology* 2(2), 195–217 (2010)
6. Di Lucca, G.A., Fasolino, A.R., Faralli, F., Carlini, U.D.: Testing Web applications. In: Proc. of the International Conference on Software Maintenance, Montreal, Canada. IEEE Computer Society (October 2002)
7. Mansour, N., Salame, M.: Data generation for path testing. *Software Quality Journal* 12, 121–136 (2004)
8. Mansour, N., Isahakian, V., Ghalayini, I.: Scatter search technique for exam timetabling. *Applied Intelligence* 34(2), 299–310 (2011)
9. Web application Testing Tools, <http://logitest.sourceforge.net/logitest/index.html>
10. Benedikt, M., Freire, J., Godefroid, P.: VeriWeb: Automatically Testing Dynamic Web Sites, <http://www2002.org/CDROM/alternate/654/>
11. Elbaum, S., Rothermel, G., Karre, S., Fisher, M.: Leveraging user session data to support web application testing. *IEEE Transactions of Software Engineering* 31, 187–202 (2005)
12. Fejes, B.: TestWeb applications with HttpUnit, <http://www.javaworld.com/javaworld/jw-04-2004/jw-0419-httpunit.html>

13. Liu, C., Kung, D., Hsia, P., Hsu, C.: Structural testing of web applications. In: Proceedings of the 11th IEEE International Symposium on Software Reliability Engineering, pp. 84–96 (October 2000)
14. Ricca, F., Tonella, P.: Analysis and testing of web applications. In: Proceedings of the International Conference on Software Engineering, pp. 25–34 (May 2001)
15. Marchetto, A., Tonella, P., Ricca, F.: Search-Based Testing of AjaxWeb Applications. In: Proc. of 1st International Symposium on IEEE Search Based Software Engineering (May 2009)
16. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. *Science* 220, 671–680 (1983)



# Stochastic Optimisation in Computational Engineering Design

Timoleon Kipouros

**Abstract.** It is evident that the requirements and specifications for engineering products, as well as the demand for these products, have increased substantially over the last couple of decades. As a result, various engineering design tasks have become considerably more complex. These observations and facts have necessitated the development of new design approaches that offer alternatives to the traditional ways of exploring design spaces and performing engineering design. At the same time, the Mathematical Sciences have produced a number of advanced search and optimisation algorithms that can explore and assess challenging and complicated models and functions. Furthermore, significant advances have been made in the field of Information Technology in terms of utilising massively parallel computational power. It has been shown that all these advancements can be exploited in complementary and synergistic ways when combined appropriately, producing a complete computational engineering design system. In this paper, a guide for deploying all these available technologies in efficient and appropriate ways is presented, illustrated with applications to real-world engineering problems in which not only are innovative solutions produced but also previously unidentified avenues of research are revealed.

## 1 Introduction

There is a seemingly constant stream of remarkable advances in many scientific disciplines, including computer gaming and geometry management, flow and performance evaluation tools from the aeronautical and aerospace engineering disciplines,

---

Timoleon Kipouros

University of Cambridge, Department of Engineering, Trumpington Street,  
Cambridge CB2 1PZ, UK

e-mail: [tk291@eng.cam.ac.uk](mailto:tk291@eng.cam.ac.uk)

Cranfield University, School of Engineering, Department of Power and Propulsion,  
Cranfield MK43 0AL, UK

e-mail: [t.kipouros@cranfield.ac.uk](mailto:t.kipouros@cranfield.ac.uk)

optimisation algorithms from mathematical and computing sciences, and visualisation techniques for multi-dimensional data. All these technologies can be utilised in the context of computational engineering design.

Furthermore, the technical specifications in aviation for future aircraft and jet engines have been set to very high and challenging targets, in terms of emissions, noise levels, flight paths, and overall efficiency [5].

These industrial design objectives reflect directly to the energy and power and propulsion sciences, where these applications can extend from fine geometrical modifications of turbomachinery blade components or aircraft wings to whole engine and whole aircraft conceptual and detailed design. The primary objective is always to improve the efficiency of these machines, and at the same time to achieve more sustainable operation within the environmental conditions, and satisfy strict regulations and constraints that combine aerodynamic, structural, aeroacoustical, cost, or manufacturing considerations.

The complexity of optimisation problems increases dramatically when the number of design variables and objective functions increases. In addition, the different hard and soft constraints that must be considered in order to maintain the practicality of new design configurations test the capabilities of the optimisation algorithms to their limits.

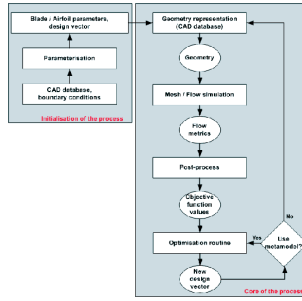
Hence, computational design methodologies are essential in contemporary engineering. Furthermore, new methodologies and technologies that could not have been investigated before can be explored with these design tools and systems.

## 2 Computational Design Approach

Figure 1 describes the iterative design process within a computational environment. Even though it relates to shape optimisation examples, the methodology is generic and can be applied to any engineering design problem. The initialisation of the process happens when the detailed CAD (Computer Aided Design) definition of a geometry is parameterised and expressed through a manageable string of numbers, the design vector. This procedure is executed only once, at the beginning of the whole operation. In general cases, this stage of the process expresses the translation of the design problem of interest through the design parameters.

Once this step is complete, the main loop of the process is executed, where the geometry is translated back into a CAD definition, in order for the computational mesh to be generated, which in turn enables the computational flow evaluation using a CFD (Computational Fluid Dynamics) tool. This process can, of course, be replaced by other appropriate evaluation packages according to the type of design problem under consideration, for example: FEA (Finite Element Analysis) for structural analysis, an analytical model for thermal analysis, or combustion CFD modelling for combustion metrics evaluation.

The process continues by assessing the flow metrics and constraints, as well as any geometrical characteristics, which describe the objective functions of the design



**Fig. 1** The Computational Engineering Design Cycle supported by surrogate modelling

problem. The evaluated target functions, together with the corresponding design vectors, are provided to the core of the whole process, the optimisation algorithm. At this stage, the optimisation algorithm performs intelligent operations that produce the new candidate design descriptions of the problem that require evaluation and assessment, and hence the workflow closes the loop. Additional information and details can be found in [19, 34].

However, rather than executing the computationally expensive tasks that evaluate the objective metrics of interest, a surrogate model can be deployed that can provide a prediction of the objective function values to the optimisation routine. As a result, the whole design process can be accelerated significantly.

It should be emphasised here that each of these computational tools has to exhibit particular characteristics that enable them to be applied in computational engineering design. The geometry management tool should be flexible enough to allow the exploration of innovative design configurations, but at the same time should secure high accuracy and ideally utilise a small number of design parameters. The evaluation tool should always be sufficiently accurate that it at least predicts the correct trends in the objective metrics and does not provide misleading information to the optimisation routine. Finally, the design space exploration should be performed by an application-specific optimisation algorithm, which is properly tuned.

In the following sections, detailed descriptions of these characteristics and of the methodology to deploy metamodels within the computational engineering design cycle are presented, with examples of real-world design applications.

### 3 Indispensable Components of Automatic Design Systems

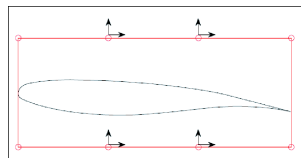
As presented earlier in Section 2 in order to build an automatic computational design system for engineering applications, three main components are required. These tools perform the tasks of geometry management, evaluation of the objective functions, and exploration of the design space. In addition, there are ancillary tools

that can be deployed within the design cycle to enhance the practicality and viability of computational design approaches for real-world applications. These tools are metamodels and advanced Grid computing architectures.

### 3.1 *Parameterisation of the Design Problem*

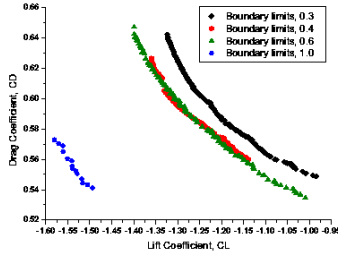
When shape optimisation problems are considered, the ability to efficiently manage geometrical modifications is a vital characteristic for the whole design system. Not only should enough flexibility be achievable, but also a low number of design parameters should be required. However, this is a general comment that applies to any type of real-world engineering design case. There are a number of techniques that can efficiently manage different types of geometries, and an excellent review of these methods is presented in [32].

Flexibility is essential in order to allow the search exploration method to investigate innovative areas that can produce significant improvements in the quality of the final optimum solutions. As an example, a benchmark 2D airfoil aerodynamic case has been considered for demonstration of these characteristics. The changes to the airfoil section are managed using the Free Form Deformation (FFD) technique [33] and the 8 design parameters of the problem express the horizontal and vertical displacements of the control points of the FFD lattice, as presented in Fig. 2. The two objective functions express the lift and drag aerodynamic coefficients subject to two hard geometric constraints on the thickness of the airfoil at 25% and 50% of the chord. This ensures that the thickness of the new optimum designs should not drop below the datum reference values, in order to secure sufficient space for the two main spars that support the wing of an airplane.



**Fig. 2** The parameterisation of the 2D airfoil section using Free Form Deformation

In Fig. 3 the effectiveness of a wide design space is clearly demonstrated. The boundary limits correspond to the range of variability of the FFD control points in both directions (positive and negative). As these ranges increase, more space for improvement develops that the optimisation algorithm can explore. Initially, there is improvement to the optimum level achievable and then the spread of the Pareto Set increases as well. Finally, the 'global' optimum area may be revealed with the highest values of variability of the design parameters. The Multi-Objective Tabu Search (MOTS) algorithm [18] has been employed for the investigation of this test case.



**Fig. 3** The significance of the degree of flexibility to the design space for a 2D shape optimisation test case

### 3.2 Evaluation and Modelling of the Objective Functions

In engineering design problems there are usually many conflicting metrics that express the overall efficiency of a machine or device. Hence, it is important the designer is able to identify the critical features that should be improved, but at the same time subject to the appropriate set of constraints that give significance and practicality to the design problem.

In all cases, the evaluation tool(s) should be accurate enough and able to capture the physics of the objective and constraint metrics at as a low computational cost as possible. At all times though, the trends and the overall behaviour of the physical objective metrics should be captured correctly, in order to avoid providing any misleading information to the optimisation algorithm.

As an example, for the detailed aerodynamic design of axial compressor blade rows the flow separation and the general and secondary factors of losses that develop over and along the span of the blade should be considered for simultaneous improvement in order eventually to achieve a highly efficient aero-engine. The four objective metrics are blockage, entropy generation rate, profile losses and endwall losses – detailed definitions and descriptions of the modelling of these metrics can be found in [24]. In addition, the mass flow rate is treated as an equality constraint, and the mass averaged flow turning, the minimum radius of the leading edge, and the tip clearance of the blade are considered as inequality constraints. The objective functions, equation 1, are normalised and include penalty function terms to handle the aerodynamic and geometric constraints.

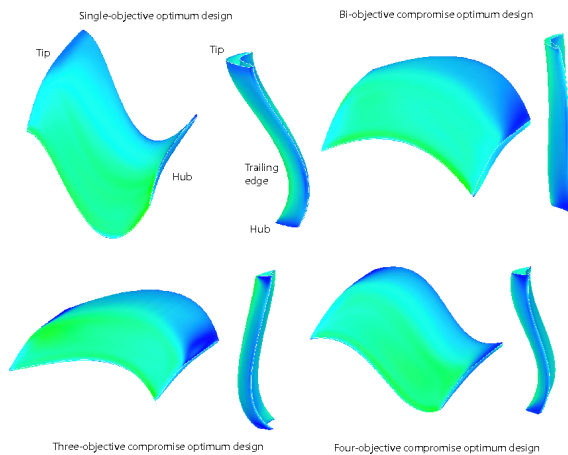
$$\begin{aligned}
 f_i = & \frac{M_i}{M_{i,0}} + 250 \left( 1 - \frac{\dot{m}}{\dot{m}_0} \right)^2 + 0.4 \max^2 \left( 0, 1 - \frac{R_{LE}}{R_{LE,0}} \right) \\
 & + 500 \max^2 \left( 0, 1 - \frac{\Delta\theta}{\Delta\theta_0} \right) + 0.5 \max^2 \left( 0, 1 - \frac{C}{C_{lim}} \right)
 \end{aligned}
 \tag{1}$$

In equation (1)  $M$  represents any of the objective metrics,  $\dot{m}$  is the mass flow rate,  $R_{LE}$  is the minimum radius of the leading edge of the blade,  $\Delta\theta$  is the mass-averaged flow turning, and  $C$  measures the tip clearance of the blade. The zero subscripts identify the equivalent quantities for the datum blade geometry, the initial design in the optimisation.

The treatment of the mass flow rate as an equality constraint is essential, in order to secure applicability of the improved design configuration to the same compressor core of the aero-engine, which is designed to operate within specific limits. Furthermore, the constraint to the leading edge radius is imposed in order to avoid sharp blades that will express very poor off-design performance. The tip clearance constraint expresses manufacturing proximity considerations that secures enough thickness of the compressor blade at the tip, and hence allows the screw to be fixed and the blade to be installed around the annulus of the compressor core. Finally, the mass-averaged flow turning secures appropriate inlet conditions to the subsequent blade row.

The geometry is parameterised using a Partial Differential Equation approach (3), giving a compact but flexible representation of the design in a design vector consisting of 26 variables. It should be noted here that these parameters do not reflect direct geometric characteristics (thickness, chord length, etc.) or any other engineering properties.

The dimensionality of the optimisation cases should be treated carefully, as it dramatically increases the complexity and computational cost required to fully explore the design problem. However, depending on the complexities of the design problem each time, it may be necessary to consider more than two objective functions in order to fully manage all the critical physical aspects that control the overall optimal behaviour of a machine.



**Fig. 4** Optimal blade geometries found by different optimisation approaches

For this particular example, initially a single-objective study was performed by Harvey *et al.* [15], considering blockage as the only objective function, subject to the same set of constraints as expressed in Eq. 1. Next, a bi-objective optimisation case, considering blockage and entropy generation rate as objectives, was investigated [25] leading to a three-objective case, with profile and endwall losses and blockage as the objectives [24]. Finally, the investigation has been extended to a four-objective case in which blockage, profile and endwall losses, and entropy generation rate are all to be minimised.

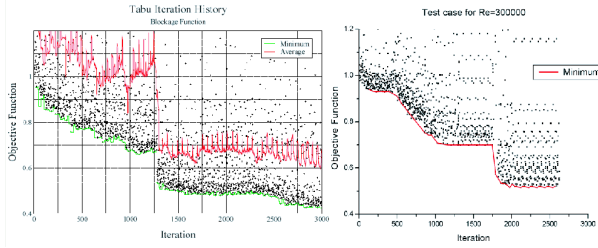
The richness of the design space for this application is readily apparent from the variation in the geometric characteristics of the various optimum designs, as appear in Fig. 4. However, the optimum design from the four-objective study appears to combine and blend geometric characteristics identified in optimum designs from each of the previous studies. This observation suggests that a complete investigation of this particular design problem has now been achieved.

### 3.3 Search Exploration of the Design Space

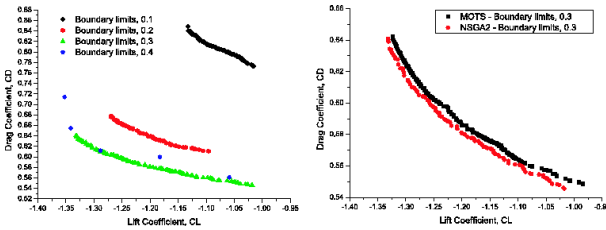
The optimisation operation within the computational engineering design cycle is the core of the whole process. However, not only the appropriate selection of optimisation technique should be considered, given the morphology and characteristics of the design space under investigation, but also the modelling and tuning of the optimisation algorithm is equally essential for the successful execution and exploration of an engineering design problem.

For the design problem described in Section 3.2 a number of detailed investigations of optimisation scenarios have been developed. For the single-objective optimisation, two complete simulations have been executed and presented in Fig. 5. Even though the same single-objective Tabu Search algorithm has been employed, the final optimal result is different. Apparently, during the second execution, shown on the right hand side of Fig. 5, the optimiser has been trapped in a local optimum area, as the pattern that has developed of design points visited reveals. This is a result of inappropriate tuning of the Tabu Search optimisation algorithm. The correct modelling of an optimisation algorithm primarily depends on the number of design parameters and objective functions.

Furthermore, the benchmark aerodynamic design case described in Section 3.1 has also been used to highlight the importance and significance of suitable selection of the optimisation algorithm. Figure 6 illustrates the optimum level achievable for gradually increased ranges of variability of the design parameters when using the NSGA-II algorithm [6]. It is again quite clear that with a larger design space, better exploration of the design problem can be attained. However, when the geometric flexibility exceeds a threshold the NSGA-II algorithm struggles to satisfy the hard geometric constraints and cannot achieve any progress in the exploration of the design space (left hand side in Fig. 6). As a result, the technique cannot reveal the 'global' optimum area as presented in Fig. 3. Apart from that, a random generation of the initial population contains a high fraction of aerodynamically infeasible



**Fig. 5** The importance of the optimisation modelling in engineering design for a 3D shape aerodynamic design of compressor blades



**Fig. 6** The importance of the appropriate optimisation algorithm selection for a 2D shape aerodynamic design of airfoil profiles (MOTS vs. NSGA-II)

designs, and hence it is very difficult for any Genetic Algorithm-type optimisation method to perform competitively in aerodynamic shape optimisation cases.

Figure 6 (right hand side) also highlights the fact that for the same extent of the design parameter boundary limits, the population based algorithm benefits from its global search characteristics and produces a trade-off that fully dominates the one produced with a Tabu Search exploration. However, it has to be noted that the optimisation level between the two Pareto fronts is very similar and, quite remarkably, the diversity of the trade-offs is very similar as well. There is only an issue with the richness of the final Pareto front for all the traditionally-defined, population-based algorithms as, implicitly, a maximum size is set once the number of individuals within a population is fixed. In contrast, there is no limit assigned to the size of the Pareto front in MOTS.

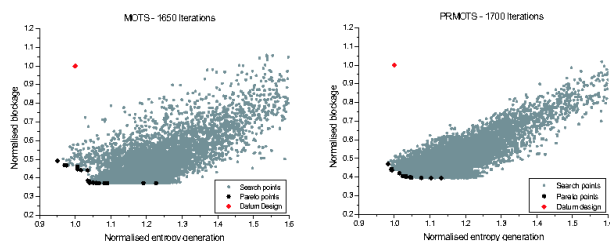
### 4 Contributions of Optimisation Intelligence

Practical engineering design problems often require a large number of design parameters in order to model design geometry and configuration accurately and flexibly enough. As consideration of multiple objectives inevitably increases the proportion



of the design space in which optimal solutions are to be found, reducing the dimensionality of the design space can significantly improve the tractability of multi-objective design optimisation problems. In order to achieve this, sophisticated parameter selection mechanisms are required.

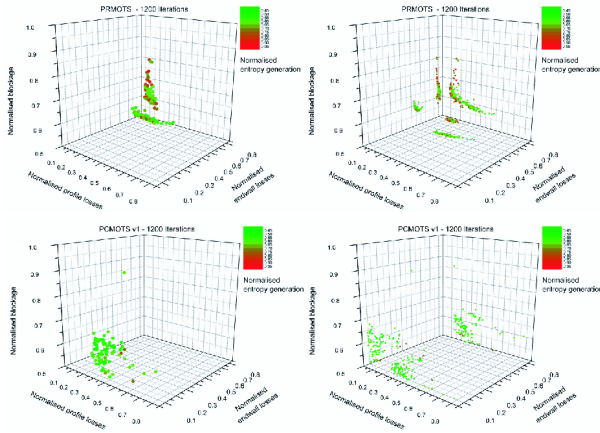
A variant of the Multi-Objective Tabu Search algorithm has been developed (PRMOTS) ([18, 21]) facilitating a sophisticated system for selecting the design variables to be modified in the search, based on the work of [20]. This seeks to identify the variables that have the greatest impact on the performance of the current design, based on geometric dissimilarity, and prioritises them at each local search iteration, so that the variables to be modified are chosen on this basis rather than randomly. The prioritisation procedure occurs at specific, user-defined intervals, and this strategy is based on the idea of *path relinking*, proposed by Glover [14].



**Fig. 7** Demonstration of the efficacy of the advanced parameter selection technique based on Path Relinking (MOTS vs. PRMOTS)

Figure 7 highlights a more consistent design space exploration towards the 'global' optimum area for the PRMOTS algorithm, when applied to the bi-objective investigation of the 3D shape aerodynamic design problem that is described in Section 3.2. More importantly, the computational cost has been reduced by 50%, when compared with the application of the MOTS algorithm, and the rate of aerodynamically feasible solutions has been increased from 17%, for the MOTS algorithm, to 48% for the PRMOTS. In addition, a smoother Pareto front is revealed with better exploration of the compromise design area. The importance of this optimum region should be emphasised here, where the design configurations balance all the objective metrics at an optimal level, and hence the optimum design configurations behave robustly over a range of critical flow metrics.

Another variant of the MOTS algorithm, PCMOTS, has been developed and presented by Ghisu *et al.* [11], who applied it successfully to a real-world aerodynamic test case, the preliminary design optimisation of the compressor core of an aero-engine. This implementation is based on Principal Component Analysis, and the aim is to identify an optimal reorientation of the design parameters and to prioritise the most energetic of these rotated design parameters. The result is a more efficient search process that requires fewer design evaluations to find high quality solutions, since a reduction in the dimensionality of the design space is achieved.



**Fig. 8** Demonstration of the efficacy of the advanced parameter selection technique based on Principal Component Analysis (PRMOTS vs. PCMOTS)

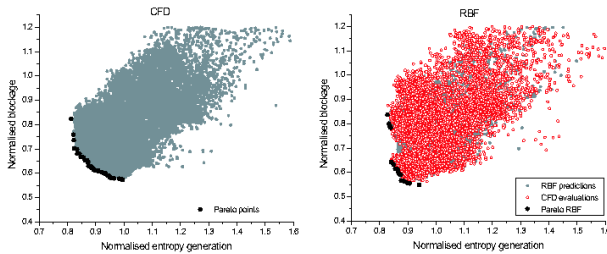
The behaviour of PCMOTS in the same 3D aerodynamic design case considering all the four objective functions was similar to that already shown by Ghisu *et al.* [11, 12, 13] for the preliminary design of the compressor core of an aero-engine. The use of this technique significantly reduced the dimensionality of the design space, improving at the same time the quality of the Pareto front. Additionally, the number of evaluations of candidate designs required per optimisation step reduced from 25 (for PRMOTS) to 18, further reducing the computational cost. In particular, considerably reduced values of normalised endwall losses were found; the values of normalised blockage and entropy generation were reduced noticeably, while the region of minimum normalised profile losses was perhaps not similarly well explored. Figure 8 presents the Pareto fronts found with PRMOTS and PCMOTS and the differences are clear, especially for the normalised endwall losses.

## 5 Auxiliary Tools

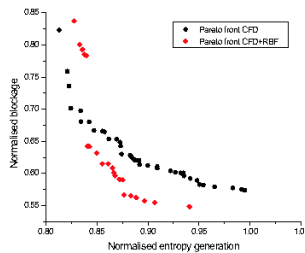
When considering further reducing the computational cost required for successful and efficient exploration of complex engineering design problems, the deployment of meta-models within the computational design cycle (Fig. 1) is essential. However, the surrogate models should be utilised in such a way that assists accelerated exploration only in “downhill” design areas. When a new candidate geometry is significantly distant (in terms of the Mahalanobis distance) from the available designs, the prediction would probably be quite poor. Using the Reduced Order Model (ROM), built on points available in the data base, to evaluate a “distant” point would be an extrapolation, rather than an interpolation. Hence, the main challenge is not to generate a model which can completely replace the accurate, but expensive,

evaluation tool (CFD in the present example), but to define the boundaries of reliability of that model. In other words, there is a need to define a method for deciding whether the model is reliable or CFD is required. For this purpose, three main criteria are suggested:

- (i) High leverage points are excluded from the prediction. In essence this criterion rejects predictions at points whose distance from the remaining points is too large.
- (ii) For each new point P it is necessary that the variation between the predicted value of each objective function and the equivalent objective functions for the closest few points to P be less than a strictness threshold in order for the prediction to be considered reliable.
- (iii) When a new point P is predicted by the surrogate model to be a Pareto design, then it is re-evaluated by CFD, in order to eliminate any over-prediction of the objective function values.



**Fig. 9** Demonstration of the efficacy of the meta-model support for a 3D shape optimisation test case



**Fig. 10** Close comparison of the two Pareto Sets revealed with direct detailed evaluations and meta-model support

The current history of an optimisation process, all the points visited in the design space, delivers the necessary information for on-line training of the surrogate model. This process can be implemented in real time, but the ability of the meta-model to make reliable predictions depends on and defines a minimum number of input designs required for training. The meta-model cannot be deployed within the optimisation process unless this criterion is satisfied.

The approach described above is generic and can be applied for any type of surrogate model. In the present example a Radial Basis Function (RBF) [4] model is used [22].

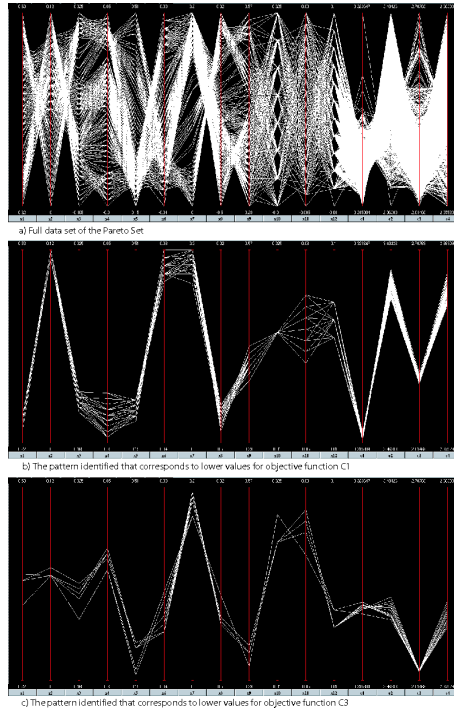
Figure 9 presents the search patterns revealed from two optimisation runs for the bi-objective investigation of the 3D shape compressor blades design case: the first used PRMOTS with CFD; the second used PRMOTS with CFD+RBF system. Similar behaviour of the two runs is apparent not only in the search patterns but also in the Pareto fronts found. Perhaps surprisingly, the Pareto front in the CFD+RBF case is somewhat more advanced than that found in the CFD-only case. This is obvious when comparing the two Pareto fronts in Fig. 10, where the compromise and right-hand end optimum areas of the CFD-only front are dominated by the front found in the CFD+RBF case. This suggests that the efficiency of the search of the design space has been improved by the use of the surrogate model. Although the left-hand end of the Pareto front found in the CFD+RBF case is not very rich, the optimum level achieved is comparable to that found in the CFD-only case. Most importantly, a 60% reduction of the overall computational cost for the exploration of this test case has been recorded.

## 6 Grid Computing

Another scientific discipline that offers complementary technology to the computational engineering design is eScience. Normally, many design systems developed by engineers exhibit powerful capabilities for tackling complex, non-linear, and discontinuous problems, but with weaknesses in the ease of use and range of applicability. On the other hand, design systems developed by computer scientists often integrate very powerful computational infrastructure, but with limited emphasis on the suitability for practical engineering problems. However, there is a platform with advanced optimisation functionalities that can efficiently manage the simultaneous execution of computational tasks through cluster and grid environments named Nimrod/O [1, 26]. Only when such capability exists can engineers consider difficult design problems that require substantial computational resources due to the size of the design space, the size of the design problem, and the computational cost for the evaluation of the critical metrics of each discipline. Such design software facilitates all levels of parallelisation, from domain decomposition to functional decomposition and multi-threaded execution of computationally intensive jobs combining cluster of clusters and utilising all the available hardware resources between institutions and industrial companies.

## 7 Post-optimisation Analyses

Successful application of these methods of computational engineering design generates vast quantities of data on potential optimal designs. To gain maximum value from the optimisation process, designers need to visualise and interpret this information leading to better understanding of the complex and multimodal relations



**Fig. 11** An example of post-optimisation analysis significance for the design of wind turbines test case. The problem is described with 8 design parameters ( $x_1$ - $x_8$ ) and 4 objective functions ( $c_1$ - $c_4$ ) comprising a 12-dimensional hyper-space

between parameters and objectives, and better decision-making between multiple and strongly conflicting criteria. It has been identified that the Parallel Coordinates interactive visualisation method has considerable potential in this regard. This methodology involves significant levels of user-interaction, making the engineering designer central to the process, rather than the passive recipient of a deluge of pre-formatted information.

The aim is to be able to visualise the whole design space of an engineering design problem in relation to the objective functions. These sets of data develop a multi-dimensional space and advanced scientific visualisation techniques are required. The suggested approach is based on parallel coordinates multidimensional geometries representation, introduced by Inselberg [17]. The parallel coordinates technique is very popular in the field of information visualisation [36]. Using this method, it is possible to visualise multivariate relations (i.e., subsets of  $R^N$ ) by mapping them uniquely into indexed subsets of  $R^2$ . This type of visualisation directly reflects the conversion of the multidimensional design vector of an optimisation design problem to a two-dimensional space. Hence, the complexity of analysing the high-dimensional design space is reduced significantly when we perform the same analysis in two dimensions.

One of the advantages of parallel coordinates is the uniform treatment of multiple dimensions that is paramount in exploratory tasks. Another advantage is that the technique allows users to interact with the data in many ways. This is very useful, as interaction is known to augment the knowledge acquisition process [36]. One of the most common tasks in information acquisition is cluster analysis, and parallel coordinate visualisations have been considered for this purpose in order to identify the patterns and combinations of design parameters that relate to optimum performance of specific objective functions for the aerodynamic design of wind turbines [8] as presented in Fig. 11.

Furthermore, this technique of post-optimisation analysis can be expanded and deployed while an optimisation process progresses, in a pro-active way, by identifying the importance of each design parameter and fixing the variability of those that are no longer significant. As a result, a reduction in the dimensionality of the design problem can be achieved that produces accelerated exploration of the design space [23].

## 8 Conclusions and Future Directions

In this paper a complete methodology for advanced computational engineering design practice has been presented. All the critical aspects of each module of such complicated systems have been analysed and demonstrated through real-world engineering applications: the 3D aerodynamic design of compressor blades and wind turbines, and the preliminary design of the compressor core of an aero-engine.

Additional challenging engineering design examples have been successfully investigated following these principles, as presented in [37] for the aerodynamic design of multi-element airfoils, as well as the structural design of gears in helicopter transmission systems [30]. Additionally, the successful design and initial development of micro-combustor devices is addressed in [31] and the design of bio-fuels in [28].

It has been confirmed that optimisation is an enabling technology in innovation and proven to be beneficial in exploring engineering design problems. Furthermore, this methodology leads to innovative solutions in shorter times, if at all possible, and releases time to the human designer for creative thinking. The trade-offs and design spaces are explored and revealed, which are understood by no other means. Insight into the design space and physical mechanisms responsible for improved efficiency can now be identified through visualisation and analyses of optimisation data that further improve decision making. Finally, the optimisation process is an optimisation problem itself.

As for future directions for research, there are already investigations towards the development of hybrid optimisation algorithms that combine merits of different methods ([27, 29, 10]), as well as examples of high-level interactive optimisation [16]. Additionally, application-specific optimisation methods and strategies should be investigated, explored and developed [38], as well as sophisticated mechanisms and tools that minimise the design cycle time [35, 9, 2, 7].

**Acknowledgements.** The author would like to acknowledge Dr Tiziano Ghisu, Dr Andrew Lewis, and Dr Geoffrey T. Parks for their valuable comments and proof reading of the manuscript, and Professor Mark Savill for his continuous support.

## References

1. Abramson, D., Lewis, A., Peachey, T., Fletcher, C.: An automatic design optimization tool and its application to computational fluid dynamics. In: Proceedings of Supercomputing (2001)
2. Adra, S.F., Dodd, T.J., Griffin, I.A., Fleming, P.J.: A convergence acceleration operator for multiobjective optimisation. *IEEE Transactions on Ev. Comp.* 13(4), 825–847 (2009)
3. Bloor, M.I.G., Wilson, M.J.: Efficient parametrization of generic aircraft geometry. *Journal of Aircraft* 32(6), 1269–1275 (1995)
4. Buhmann, M.D.: Radial basis functions: theory and implementations. Cambridge University Press (2003)
5. Clean sky at a glance: Bringing sustainable air transport closer (2012), <http://www.cleansky.eu/lists/documents> (cited April 17, 2012)
6. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Ev. Comp.* 6, 182–197 (2002)
7. Eftang, J.L., Huynh, D.B.P., Knezevic, D.J., Patera, A.T.: A two-step certified reduced basis method. *Journal of Scientific Computing* (2011), doi:10.1007/s1091501194942
8. Fischer, G.R., Kipouros, T., Savill, A.M.: Multi-objective shape optimisation for horizontal-axis wind turbine blades. AIAA-2012-1353 (2012)
9. Forrester, A.I.J., Sobester, A., Keane, A.J.: Engineering design via surrogate modelling: A practical guide. John-Wiley and Sons, Chichester (2008)
10. Ghiasi, H., Pasini, D., Lessard, L.: A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems. *Eng. Opt.* 43(1), 39–59 (2011)
11. Ghisu, T., Parks, G.T., Jaeggi, D.M., Jarrett, J.P., Clarkson, P.J.: The benefits of adaptive parametrization in multi-objective Tabu Search optimization. *Eng. Opt.* 42, 959–981 (2010)
12. Ghisu, T., Parks, G.T., Jarrett, J.P., Clarkson, P.J.: An integrated system for the aerodynamic design of compression systems - Part I: Development. *ASME Journal of Turbomachinery* 133(1), 011011–1–011011–10 (2011)
13. Ghisu, T., Parks, G.T., Jarrett, J.P., Clarkson, P.J.: An integrated system for the aerodynamic design of compression systems - Part II: Application. *ASME Journal of Turbomachinery* 133(1), 011012–1–011012–8 (2011)
14. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Boston (1997)
15. Harvey, S.A., Dawes, W.N., Gallimore, S.J.: An automatic design optimisation system for axial compressors, Part I: Software development. *ASME GT2003-38115* (2003)
16. Hettenhausen, J., Lewis, A., Mostaghim, S.: Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Eng. Opt.* 42(2), 119–139 (2010)
17. Inselberg, A.: Parallel Coordinates: Visual multidimensional geometry and its applications. Springer, New York (2009)
18. Jaeggi, D.M., Parks, G.T., Kipouros, T., Clarkson, P.J.: The development of a multi-objective Tabu Search algorithm for continuous optimisation problems. *European Journal of Operational Research* 185, 1192–1212 (2008)
19. Keane, A.J., Nair, P.B.: Computational approaches for aerospace design: The pursuit of excellence. John-Wiley and Sons, Chichester (2005)

20. Kellar, W.P.: Geometry modelling in computational fluid dynamics and design optimisation. PhD Thesis, Cambridge University, Department of Engineering (2003)
21. Kipouros, T., Jaeggi, D.M., Dawes, W.N., Parks, G.T., Savill, A.M.: Multi-criteria optimisation of turbomachinery blades: investigating the trade-off surface. AIAA-2005-4023 (2005)
22. Kipouros, T., Molinari, M., Dawes, W.N., Parks, G.T., Savill, A.M., Jenkins, K.W.: An investigation of the potential for enhancing the computational turbomachinery design cycle using surrogate models and high performance parallelisation. ASME GT2007-28106 (2007)
23. Kipouros, T., Ghisu, T., Parks, G.T., Savill, A.M.: Using post-analyses of optimisation processes as an active computational design tool. ICCES 7(4), 151–157 (2008)
24. Kipouros, T., Jaeggi, D.M., Dawes, W.N., Parks, G.T., Savill, A.M., Clarkson, P.J.: Insight into high-quality aerodynamic design spaces through multi-objective optimization. CMES 37(1), 1–23 (2008)
25. Kipouros, T., Jaeggi, D.M., Dawes, W.N., Parks, G.T., Savill, A.M., Clarkson, P.J.: Biobjective design optimization for axial compressors using Tabu Search. AIAA Journal 46(3), 701–711 (2008)
26. Kipouros, T., Peachey, T., Abramson, D., Savill, A.M.: Enhancing and developing the practical optimisation capabilities and intelligence of automatic design software. AIAA-2012-1677 (2012)
27. Martínez, S.Z., Montaña, A.A., Coello Coello, C.A.: A nonlinear simplex search approach for multi-objective optimization. IEEE Congress on Ev. Comp., 2367–2374 (2011)
28. Mazlan, N.M., Savill, A.M., Kipouros, T., Li, Y.-G.: A numerical study into the effects of bio-synthetic paraffinic kerosine blends with jet-A fuel for civil aircraft engine. ASME GT2012-68754 (2012)
29. Molina-Cristóbal, A., Palmer, P.R., Skinner, B.A., Parks, G.T.: Multi-fidelity simulation modelling in optimization of a submarine propulsion system. In: IEEE Vehicle Power and Propulsion Conference (2011)
30. Pandya, B., D'Souza, N., Kipouros, T., Savill, A.M.: Structural design optimisation for helical gear pairs. In: NAFEMS-UK Conference, Engineering Simulation: Contributing to Business Success (2010)
31. Saddawi, S.D., Kipouros, T., Savill, A.M.: Computational engineering design for micro-scale combustors. ASME GT2012-69522 (2012)
32. Samareh, J.A.: Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. AIAA Journal 39(5), 877–884 (2001)
33. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. SIGGRAPH 20(4), 151–160 (1986)
34. Shahpar, S.: Automatic aerodynamic design optimisation of turbomachinery components - An industrial perspective. VKI Lecture series on Optimisation methods and tools for multicriteria/multidisciplinary design, pp. 1–40 (2004)
35. Shahpar, S.: Design of experiment, screening and response surface modelling to minimise the design cycle time. VKI Lecture series on Optimisation methods and tools for multicriteria/multidisciplinary design, pp. 1–49 (2004)
36. Siirtola, H., Rähkä, K.-J.: Interacting with parallel coordinates. Interacting with Computers 18(6), 1278–1309 (2006)
37. Trapani, G., Kipouros, T., Savill, A.M.: Computational aerodynamic design for 2D high-lift airfoil configurations. In: Pegasus AIAA (2010)
38. Van den Braembussche, R.A.: Tuning on optimization strategies. NATO RTO-EN-AVT 167 (2011)



# Evolutionary Algorithms and Water Resources Optimization

Oluwatosin Olofintoye, Josiah Adeyemo\*, and Fred Otieno

**Abstract.** Heuristic optimization models with varying degree of complexity have been widely applied for resolving water resources optimization and allocation problems. Nevertheless, there still exist uncertainties about finding a generally consistent and trustworthy method that can find solutions, which are really close to the global optimum in all circumstances. This paper makes a review of some of the numerous evolutionary optimization algorithms available to water resources planners and managers. Evolutionary algorithms have been found propitious and useful in facilitating critical water management decisions and are becoming promising global optimization tools for major real world applications. Further research aimed at developing optimization models for water resources planning, management and optimization is therefore necessary.

**Keywords:** heuristic, optimization, global optimum, evolutionary algorithm, water resources.

## 1 Introduction

Water is a natural resource that is essential for good health and the survival of all kinds of lives on earth. Less than 1% of the water of the earth is available as fresh water on land. The rest is contained either in the oceans or in form of frozen ice on mountain tops and glaciers [1]. Under pressure from population explosion, urbanization, extravagant lifestyles, climate change, intensive agriculture and industrialization, water is fast becoming a scarce resource. This is evident from the fact

---

Oluwatosin Olofintoye · Josiah Adeyemo  
Durban University of Technology, Durban, South Africa  
e-mail: [tinmodel@yahoo.com](mailto:tinmodel@yahoo.com), [josiaha@dut.ac.za](mailto:josiaha@dut.ac.za)

Fred Otieno  
Durban University of Technology, Durban, South Africa

\* Corresponding author.

that a lack of water to meet daily needs is a reality for one in three people around the world today. Health consequences of water scarcity and its impact on daily life pose a threat to national growth and impede international development [2]. It is absolutely necessary therefore, to sensitize people of all nations about the imminent danger posed by mismanagement, which may result in the depletion of limited fresh water resources and the impact this will have on people and the ecosystems on which they depend.

From the earliest times, water resources management and allocation had been on the basis of social criteria, maintaining the community by ensuring that water is available for human consumption, sanitation and food production [3]. In some cases, there had existed rigid water right policy in which water was allocated to users according to their rights without taking into account economic efficiency in water use [4]. However, with the trends in population growth and its attributes and continuous pollution of the available water sources, there has been increased pressure on the available water resource resulting in increased conflict over its allocation and a further stress on this resource leading to scarcity [5]. In some cases, existing hydrologic policies for resolving deficit often aims at increasing water resources through the construction of more hydraulic regulation or retention works, mainly large dams. Societies have invested capital in infrastructure to maintain this allocation. Yet social changes, further population growth and water pollution has made water scarcity more widespread than ever before. Thus inefficient use of water, poor cost recovery for operating and maintenance expenses, the mounting cost of developing new water sources and problems with the quality of service in agency-managed systems has led to a search for alternatives that make water allocation and management more efficient [3]. This has oftentimes led to amendments in existing water management policies.

Water inadequacy in most countries calls for concern in the management of existing facilities since the building of new facilities requires very high investment [6]. If available water resources are not utilized efficiently and effectively, water demand may exceed available supply ultimately leading to artificial drought situations in several places on the globe in the near future. Therefore, employing advanced water use forecasting and optimization models that integrates resource planning of available supply and the simultaneous management of various demands or allocation is of paramount importance in the field of water resources planning and management [1].

Awareness of increasing water scarcity has driven efforts to model global water resources for improved insight into water resources infrastructure and management strategies [7-10]. Developing strategies that facilitate the efficient use of available water resources have been the subject of many studies in the field of water resources planning and management [4, 11-14]. Several optimization techniques that attempt to propound ways of mitigating or resolving water resources allocation problems have been reported in several studies [15-17]. In recent times, methods of evolutionary algorithms (EAs) have found widespread use in the fields of water resources single and multi-objective optimization due to their robustness in the resolution of such problems [18-20]. EAs are population-based metaheuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and

survival of the fittest in order to refine a set of solution candidates iteratively [21]. EAs often perform well approximating solutions to all types of problems because they ideally do not make any assumption about the underlying fitness landscape. Apart from their use as mathematical optimizers, EAs have also been used as an experimental framework within which to validate theories about biological evolution and natural selection, particularly through work in the field of artificial life. EAs belong to a class of search methods with remarkable balance between exploitation of the best solutions and exploration of the search space. They combine elements of directed and stochastic search and, therefore, are more robust than existing directed search methods, providing the global optimum without being trapped in local optima. Additionally, they may be easily tailored to the specific application of interest, taking into account the special characteristics of the problem under consideration and they can be easily parallelized [22]. In the past decades, several EAs that mimic biological entities behaviour and evolution have emerged. Available EAs include but are not limited to genetic algorithm (GA), differential evolution (DE), evolution strategy (ES) and genetic programming (GP). The superiority of EAs in the solution of single and multi-objective optimization problems over other optimization techniques have been demonstrated by several researchers in recent years [23-25].

## 2 Water Resources Management Using Genetic Algorithms

Genetic algorithms are a subclass of EAs where the elements of the search space are encoded using binary strings or arrays of other elementary types, although versions of GAs that employ the use of real valued parameters are now available [21]. In a GA setup, possible solutions of a mathematical problem are represented by strings of genetic factors called chromosomes. A set of individuals makes up a generation. The generation is allowed to evolve through the genetic operations of selection, crossover and mutation till a global optimum solution is obtained [26]. The successful application of GA in the solution of water resources management problems has been reported in several studies [27-29]. [30] presented an approach of GA optimization (GAO) for generating optimal water network topology (OWNT) with an objective to minimize total low-level contaminated water (LLCW) consumption. The algorithm was applied to water systems with multiple pollutants and several LLCW sources, considering only one resource at a time and complying with all restrictions. The OWNT was viewed as an oriented graph, starting from unit operations with the lowest level of contaminants at entrance using an ordering rule based either on the load or the maximum LLCW consumption. A mathematical model based on the total and contaminant species mass balances, together with the input and output units' constraints was developed and solved using the GAO. Each internal flow defined the genes on a chromosomal representation of the decision variable. The individual chromosomes were interbred according to their frequency of selection, using one-point crossover method, then mutation was applied to randomly selected members of the new population. Comparison of the results with a method of mathematical programming showed that the performance of the GAO

was satisfactory, thus the GAO could be used as an alternative for minimizing total LLCW consumption in water network topologies.

It has been observed that water quality of most large rivers in South Korea is poor due to industrialization and pollution due to the high population resident in the cities [26]. Also, seasonal variation in river flow is very large. In the drought season, low flows lead to an increase in the pollution level. Pollution is a serious problem in the middle and lower parts of the rivers because many industrial facilities and large populated cities are located around them. It was especially noted that in the Youngsan river basin, one of the most polluted rivers in South Korea, water quality has decreased due to heavy pollutant loads from Kwangju city and surrounding areas. In the past, wastewater treatment policy for polluted rivers in Korea has been to construct secondary treatment plants for untreated areas, and secondly, to construct advanced treatment plants for the river sections whose water quality is impaired. Unfortunately, the water quality goal of the ministry of environment was not met. Thus, [26], in a study to achieve water quality goals and wastewater treatment cost optimisation in the affected river basin developed a mathematical water quality management model integrating a GA. In the study, total wastewater treatment costs in the basin were calculated as the sum of the treatment cost for each plant, which was based on the treatment type. Treatment cost for a wastewater treatment plant is the sum of the annual repayment for capital construction costs and annual operation-and-management costs. In calculating the annual repayment for capital costs, the construction costs of already completed plants were not included, but the construction costs for the new plants and the additional costs for the capacity expansion and the upgrade of existing plants were included. An amortization period of 20 years and the social discount rate of 11.3% were used in the study. Pollution source, land use, geographic features and measured water quality data of the river basin were incorporated into the Arc/View geographic information system database to facilitate the selection of treatment type and computation of treatment cost for 26 wastewater treatment plants in the river basin. In applying the algorithm to the water quality management, the chromosome was designed in such a way that the genes represented the treatment level at each wastewater treatment plant. A careful selection of proper genetic operators among various kinds of genetic operators was made. The chromosomes were encoded using a system of binary digits. In the optimisation process of the GA, a population of 90 chromosomes was used; crossover and mutation probabilities were set at 80% and 1% respectively. Algorithm was repeated for 100 generations. The fitness of the chromosomes were evaluated using the results of the forecasts of water quality and treatment cost. The fitness value was represented as the inverse of the sum of the total treatment cost and the penalty. A penalty was given whenever the water quality goal is violated and subsequently, the fitness value is modified by linear scaling. Linear scaling was used for the proper selection of the parents. The modified fitness value of each chromosome was used for the selected parents that will produce offspring for the next generation. Results from four scenarios that do not use the GA were compared with the results of the management model using the GA. It was found that the results based on the GA were much better than those for the other four scenarios from the viewpoint of the

achievement of water quality goals and cost optimisation. It was concluded that GA could be an alternative to other mathematical programming methods, which have been applied in regional wastewater treatment cost optimisation.

[31] proposed the use of a hybrid technique based on an improved GA, to solve the optimisation problem of finding the minimum water supply concomitantly with the water network topology ensuring the maximum water reuse. The hybrid character of the algorithm is given by the local reshape of the chromosome at the gene level to cope with the mass balance restrictions, while the improvement of the GA is given by the shrinking neighbourhood cloning strategy, which favours the individuals surrounding the best-so-far solution. The GA optimisation uses each network's internal flow as a gene, assembling the topology into a chromosome. The restrictions in terms of minimum and maximum allowable flows for each gene are dealt with naturally during the population generation, simply rejecting those genes outside their feasible domain. The individuals were interbred according to their frequency of selection, using the one-point crossover method and then mutations were applied to randomly selected individuals. The algorithm was tested on test problems from literature and was proven to have better-solving capacity after which it was used for the more difficult problem of finding an optimal topology of the wastewater network when multiple contaminated water sources are available. A mathematical model describing each unit operation based on total and contaminant species' mass balances, together with input and output constraints was developed and solved using the algorithm. The topology of the wastewater network was encoded by an oriented graph, which was adequately represented by an upper triangular matrix in which the units lay on the diagonal positions. Finally, the algorithm was encoded in an easy to use software, which facilitates the computation of the optimal topology of the unit operations' network and the minimum supply water consumption, observing the imposed inlet and outlet constraints.

In a study employing the use of an enhanced GA for bi-objective pump scheduling in a water supply system, [28] argued that when a cost-effective water distribution system is to be designed, the best way is to plan a profitable schedule for a set of pump combinations. Frequently, existing infrastructures, more often than not, are not allowed to be altered or re-constructed. Thus, without modifying the infrastructure, the overall cost can still be cut down by using a good pumping schedule. Moreover, a good pump scheduling scheme is a feasible and economic way to save the pumping cost. This is because there is no need to destroy the current infrastructure and the existing pumps and pipelines still can contribute to daily water distribution. Therefore, the only thing that needs to be redesign is a good schedule, namely that, if possible, the pumps had better be operated in the night time. They further argued that in areas where groundwater is the sole water resource, it should be used temperately because depleting groundwater may cause land subsidence, an environmental change can hardly be recovered. This situation may be avoided through artificial mediation through improved pumping by pumping water intermittently. Thus by developing an eco-aware schedule, land subsidence caused by over-pumping groundwater can thus be slowed down. Therefore, an eco-aware objective was considered in the proposed model such that the final resulting schedules

can be not only cost-effective but also environmental conscious. A good pumping scheme therefore, should be a trade-off between environmental benefits and maintenance cost. In the study, a GA-based pump scheduling scheme was proposed in which pumps need to be used intermittently. In this scheme, a GA-based local search is proposed to enhance the intensification force, i.e., the exploitation of the accumulated search experience. A real-number chromosome encoding was also employed to meet the needs of the real-world problem. Two selection operators were employed in the selection procedure: one for single-objective optimization and one for bi-objective optimization. For the simple single-objective optimization (i.e. cost), the roulette wheel selection is considered while for the bi-objective optimization (i.e. cost and subsidence), a Pareto fitness ranking was used to guide the selection procedure. A single-point crossover operator was used in which a pump is randomly chosen as a crossover point and tail parts of two solutions are exchanged to produce two new offspring. Five heuristic mutation procedures were employed to access vector points in the search space. The evolutionary population size  $N$  was set at 100 and the number of generations  $G$  at 1000. The crossover rate and mutation rate used in all methods were 0.9 and 0.5, respectively. Unlike traditional GA methods, the proposed enhanced GA (EGA) has two merits. First a method of greedy selection is employed to obtain a near-optimal solution at the beginning and to accelerate the convergence speed. In comparison to conventional methods, the proposed scheme generates a higher-quality population. Therefore, only a few generations are needed to achieve desired convergence. Second, a binary local search is developed according to the properties of the problem. With this local search, each chromosome can converge at the local minimum in its neighborhood such that promising solutions are not ignored by the proposed scheme. To ease the computational tedium involved in the research, the proposed algorithm was implemented using Delphi 7, a powerful rapid application development tool. Final results showed that the schedule did not only achieve lower cost but also gained more environmental benefits. In concluding the study, [28] suggested that future researches should focus on accelerating the convergence speed of the EGA.

### 3 Water Resources Management Using Differential Evolution

Differential evolution (DE), developed by Price and Storn in 1995 [32], is a simple yet powerful heuristic method for solving nonlinear, non-differentiable and multimodal optimization problems. The algorithm combines simple arithmetic operators with the classical events of crossover, mutation and selection to evolve from a randomly generated initial trial population until a fittest solution is found. The key idea behind DE is the scheme it uses for generating trial parameter vectors. Mutation and crossover are used to generate new trial vectors while a selection scheme determines which of the vectors survives to the next generation. [32]. In recent years DE has gradually become more popular and has been used in many practical cases, mainly because it has demonstrated good robust convergence properties and is principally easy to understand [19, 33, 34]. For instance, [6] presented four strategies of an EA,

multi-objective differential evolution algorithm (MDEA) to demonstrate the potential of maximizing the farmer's total net income despite the water shortage problems in South Africa. MDEA is a novel algorithm based on the original differential evolution algorithm proposed by [35]. In the study, four strategies of MDEA namely, MDEA1, MDEA2, MDEA3 and MDEA4 were adapted to solve a multi-objective crop planning model with multiple constraints in a farmland in the Vaalharts irrigation scheme (VIS). VIS covers about 36,950 ha and is located in the east of Fhaap Plateau on the Northern Cape and North West province border in South Africa. The three objectives of the model were to minimize the total irrigation water ( $\text{m}^3$ ) and to maximize both the total net income in South African Rand (ZAR) from farming and the total agricultural output in tons. Four crops namely maize, groundnut, Lucerne and Peacan nut were planted with each crop planted in at least  $5000\text{m}^2$  of land. Monthly estimated of gross irrigation water requirements for the crops were computed following standard procedures. The volume of total irrigation water to be used for irrigating the four crops on the farm is minimized due to shortage of water in the study area. Numerical results produce quality non-dominated solutions which converged to Pareto optimal fronts. MDEA1 and MDEA2 strategies with binomial crossover method were found to be better for solving the crop planning problem presented than MDEA3 and MDEA4 strategies with exponential crossover method. It was observed that MDEA, which handles multiple constraints using a penalty function proposed by [36], runs faster with more quality Pareto optimal solutions when tested on benchmark problems. All the four strategies of MDEA namely, MDEA1, MDEA2, MDEA3 and MDEA4 found non-dominated solutions that converge to the Pareto fronts. The solutions were also diverse on the Pareto fronts. Thus, it was concluded that the four MDEA strategies are effective and robust multi-objective optimization algorithms for solving multi-objective models in water resources management especially in water deficient countries like South Africa.

According to [37], the multiple uses of the coastal water resources and the necessity of maintaining them in good quality require rational design and management. The water quality deterioration due to seawater intrusion that is observed in the coastal aquifers, especially during the summer season, has been identified as the main environmental problem for these regions and it is directly related to seawater intrusion phenomenon and its development. In an attempt to design an optimal pumping scheme in the coastal aquifer of Hersonissos in Crete, to ensure water adequacy during the summer season without enhancing the already intense seawater intrusion problem in the region, [37] understudied the optimal management of fresh water resources in coastal regions based on environmental criteria with focus on the determination of an optimal pumping scheme that will ensure adequacy of portable water demand in coastal regions without deteriorating the quality of the fresh water due to the seawater intrusion. The objective of the management model was formulated as the maximization of the total extracted fresh water volume from five pre-selected pumping locations (production wells). Constraints that ensure no further intrusion of the seawater front were imposed at ten pre-selected observation locations where the calculated hydraulic head should be greater than a specified value the end of a 10-year management period. Restrictions for all the five production wells

regarding the maximum allowable extracting rate were also set and summarized in a mathematical model. First, a simplex method was used to solve the constrained optimization problem; a piecewise linearization of the non-linear optimization problem was obtained by sequential implementation of the simplex algorithm. Secondly, the solution of the non-linear optimization problem was obtained using a Differential Evolution (DE) algorithm. In the implementation presented in the study, the constraints were combined with the objective function as penalty terms, to form a fitness function, which is minimized using the DE algorithm. A comparison of the results obtained by the two different optimization approaches was performed and a sensitivity analysis was employed in order to examine the influence of the active pumping wells in the evolution of the seawater intrusion front along the coastline. The solutions provided by the two methods were similar for values of the volume flow rates and the values of the sensitivity analyses. However a discrepancy between the two solutions was observed at a particular pumping well location where the Simplex method, contrary to the DE algorithm, provided a zero value for the corresponding volume flow rate. Additionally, as the sensitivity analysis demonstrated, the Simplex solution shows a much higher sensitivity, at the well, compared to the DE solution, which seems more robust.

#### **4 Water Resources Management Using Evolution Strategy**

Evolution strategy (ES) was developed in 1963 by Ingo Rechenberg and Hans-Paul Schwefel at the Technical University of Berlin (TUB) while solving an engineering optimization problem. ES like GA is a stochastic search heuristic based on ideas of adaptation and evolution and is conceptually similar to natural evolution. [38]. An ES is an effective continuous function optimizer, in part because it encodes parameters as floating-point numbers and manipulates them with arithmetic operators. ES primarily uses mutation and selection as search operators, which are applied iteratively. Each iteration is called a generation [32]. The use of ES as function optimizers have been reported in studies [39-41]. For example, [38] in a study aimed at finding solutions to groundwater contaminant source identification problems argued that groundwater contaminant source identification is important in environmental forensics and characterization of contamination for the purposes of regulatory enforcement and liability assessment. Further, groundwater characterization can be classified as an inverse problem, which involves the resolution of unknown system characteristics from observed data. Inverse problems are difficult to solve due to their natural ill-posedness and computational intractability. In the study, a simulation-optimization approach that couples a numerical pollutant-transport simulation model with an evolution strategy search algorithm was adopted for solution of the inverse problem. In a simulation-optimization approach, the simulation model is coupled loosely or tightly with an optimization technique to determine the model inputs that best represent the observed data. The research considered three-dimensional heterogeneous and homogeneous flow field problems with four to seven unknown parameters to be estimated, with the contaminant source located within the



aquifer. The solution method uses a loosely coupled simulation–optimization approach. In the problem context, source locations and historical contaminant release schedules were the unknowns, and were resolved from the spatially and temporally distributed observations collected at contaminant monitoring wells. The main objective was to enhance the efficiency of the simulation–optimization approach utilizing high performance technologies that minimizes the overall computation time for solving groundwater inverse problems. A parallelization approach that exploits the fine and coarse grained parallelism exhibited by simulation–optimization frameworks was employed to improve the simulation model efficiency and reduce forward model computation time. A forward model, represented by a system of partial differential equations (PDEs) was used to describe the transport processes of the groundwater system and to define the relationship between system inputs and outputs. This numerical transport model is then solved iteratively during the evolutionary search. Several variations of a groundwater source identification problem were examined and the fitness function was evaluated in terms of solution quality and computational performance. A population size of 128 vectors was used for the ES-based procedure, which was executed for 10 generations, to estimate the computation time. The results indicate that while ES performs adequately for all scenarios investigated, the performance was affected by problem complexity i.e. number of decision variables used to characterize the source. It was however found that the parallel simulation–optimization framework with the optimal configuration reduces the simulation time drastically from days to minutes when compared to a serial implementation method. The process involved in the study were computationally intensive since several hundreds to thousands of forward model evaluations are typically required for solutions to be found, hence the computational experiments were performed on the TeraGrid cluster, a mainframe computer available at the National Center for Supercomputing Applications.

## 5 Water Resources Management Using Genetic Programming

Genetic programming (GP) is a class of EA that automatically creates computer programs to perform a selected task using the principle of Darwinian natural selection. GP is a robust, dynamic and fast growing discipline and has been successfully applied and verified in the field of water resources engineering [42–44]. [45], in a study to investigate best-fit models for predicting groundwater depth in Bondville and Perry, understudied the ability of GP and adaptive neuro-fuzzy inference system (ANFIS) techniques for groundwater depth forecasting. Five GP and ANFIS models comprising various combinations of water table depth values were developed to forecast one-, two- and three-day ahead water table depths. Comparison of the accuracy of the models were made based of the root mean square errors (*RMSE*), scatter index (*SI*), Variance account for (*VAF*) and coefficient of determination ( $R^2$ ) statistics. Results showed that the GP and ANFIS models could be employed

successfully in forecasting water table depth fluctuations. However, GP was found to be superior to ANFIS in accuracy and provided explicit mathematical expressions for the problem.

A machine code-based GP for suspended sediment concentration estimation in the flow of a river at Jagual, Puerto Rico, was developed by [46]. The study proposed an application of linear genetic programming (LGP), which is an extension to GP technique, for suspended sediment concentration estimation. The authors argued that accurately estimation of suspended sediment concentration carried by a river is important with respect to channel navigability, reservoir filling, hydroelectric-equipment longevity, river aesthetics, fish habitat, scientific interests and many water resources projects. Under-estimating of sediment yield ultimately results in insufficient reservoir capacity. Hence, acquire an appropriate reservoir design and operation it is mandatory to determine sediment yield accurately. The study utilized the LGP variant of GP, which operates directly on binary machine code strings that are perturbed directly in memory and are executed directly without passing an interpreter during fitness calculation. This result in a significant speedup compared with interpreting systems. The main characteristic of LGP in comparison to tree-based GP is that expressions of a functional programming language (like LISP) are substituted by programs of an imperative language (like C). The main aim of the study was to develop an explicit formulation based on LGP to accurately estimate suspended sediment concentration. The EA adopted for the LGP applies tournament selection and puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. The loser of each tournament is replaced by a copy of the winner. In the crossover scheme adopted, a segment of random position and random length is selected in each of the two parents and exchanged. If one of the resulting children exceeds the maximum length, crossover is aborted and restarted with exchanging equally sized segments. The crossover points only occur between instructions. A mutation operation that randomly replaces instruction identifier, variables, or constants by equivalents from valid ranges was adopted. High mutation rates have been experienced to produce better results. The length of initial population of feasible programs was 64 instructions per program and the maximum number of instructions allowed per program was set to 256. LGP model was evolved for 10,000 generations. The daily streamflow and suspended sediment concentration data from two stations, Rio Valenciano Station and Quebrada Blanca Station, operated by the US Geological Survey (USGS) were used as case studies. The performance of LGP was compared with those of the adaptive neuro-fuzzy, neural networks and rating curve models employed in previous studies. Comparison of the results indicated that the LGP performs better than the neuro-fuzzy, neural networks and rating curve models based on the root mean square errors (RMSE) and determination coefficient ( $R^2$ ) statistics. Unlike the NF and ANN, which are black-box models. The LGP model presents a simple explicit mathematical formulation. It was concluded that the LGP, which is relatively much simpler than the NF and ANN can be successfully used in modeling daily suspended sediment concentrations in rivers.

[47] developed surrogate models based for evolving multi-objective management strategies for saltwater intrusion in coastal aquifers. Two different surrogate models one based on GP and the other based on modular neural network (MNN) were developed and linked to a multi-objective genetic algorithm (MOGA) to derive the optimum pumping strategy for coastal aquifer management, considering two objectives. The surrogate models were trained and tested then used to predict the salinity concentrations at different locations resulting from groundwater extraction. A two-stage training strategy was implemented for training the surrogate models. Surrogate models were initially trained with input patterns selected uniformly from the entire search space and optimal management strategies based on the model predictions were derived from the management model. Second, a search space adaptation and model retraining was performed by identifying a modified search space near the initial optimal solutions based on the relative importance of the variables in salinity prediction. The performance of the methodologies using GP and MNN based surrogate models were compared for an illustrative study area. It was found that the developed GP models had lesser uncertainty compared to MNN models and the number of parameters used in GP models was lesser than that used in the MNN models. Results also showed that GP based models were better suited for optimization using an adaptive search space.

## 6 Conclusion

Optimization models with varying degree of complexity have been widely applied for resolving water resources optimization and allocation problems. In recent times, procedures based on heuristic EAs have found wide spread applications in the fields of water resources planning, management and optimization. This paper makes a review of some of the numerous evolutionary optimization algorithms available to water resources planners and managers. Several heuristic algorithms have been applied in resolving water resources optimization problems, yet there still exist some uncertainties about finding a generally trustworthy method that can consistently find solutions that are really close to the global optimum of the problems in all circumstances [48]. The choice of optimization models is almost arbitrary as no physical basis is available to rationalize the use of any particular algorithm. Search for the proper optimization function has been the subject of several studies [49-51].

EAs have been found propitious and useful in facilitating critical water management decisions and are becoming promising global optimization tools for major real world applications. Therefore, developing single and multi-objective evolutionary optimization models that integrate resource planning of available resources and the simultaneous management of various demands or allocation is a topic that is still open for further research in the field of water resources planning and management. Further research aimed at developing evolutionary optimization procedures for water resources planning, management and optimization is therefore necessary.

## References

1. Srinivasulu, S., Jain, A.: A comparative analysis of training methods for artificial neural network rainfall-runoff models. *Applied Soft Computing* 6, 295–306 (2006)
2. WHO: 10 facts about water scarcity (2009), <http://www.who.int/about/copyright/en/>
3. Dinar, A., Rosegrant, M.W., Meinzen-Dick, R.: Water allocation mechanisms principles and examples. World Bank, Agriculture and Natural Resources Department (1997)
4. Reça, J., Roldan, J., Alcaide, M., Lopez, R., Camacho, E.: Optimisation model for water allocation in deficit irrigation systems: I. Description of the model. *Agricultural Water Management* 48, 103–116 (2001)
5. Otieno, F.A.O., Ochieng, G.M.M.: Water management tools as a means of averting a possible water scarcity in South Africa by the year 2025. In: Water Institute of South Africa (WISA) Biennial Conference (2004)
6. Adeyemo, J., Otieno, F.: Differential evolution algorithm for solving multi-objective crop planning model. *Agricultural Water Management* 97(6), 848–856 (2010)
7. Olofintoye, O.O., Sule, B.F., Salami, A.W.: Best-fit probability distribution model for peak daily rainfall of selected cities in Nigeria. *New York Science Journal* 2(3), 1–12 (2009)
8. Davies, E.G.R., Simonovic, S.P.: Global water resources modeling with an integrated model of the social-economic-environmental system. *Advances in Water Resources* 34, 684–700 (2011)
9. Olofintoye, O.O., Adeyemo, J.A.: The role of global warming in the reservoir storage drop at Kainji dam in Nigeria. *International Journal of the Physical Sciences* 6(19), 4614–4620 (2011)
10. Olofintoye, O.O., Salami, A.W.: Development and Assessment of a Quintic Polynomial Model for the Prediction of Maximum Daily Rainfall in Ilorin, Nigeria. *NSE Technical Transaction A Technical Publication of the Nigerian Society of Engineers* 46(2), 81–91 (2011)
11. Sniedovich, M.: Dynamic programming and the principle of optimality: A systematic approach. *Advances in Water Resources* 1(4), 183–190 (1978)
12. Shangguan, Z., Shao, M., Horton, R., Lei, T., Qin, L., Ma, J.: A model for regional optimal allocation of irrigation water resources under deficit irrigation and its applications. *Agricultural Water Management* 52, 139–154 (2002)
13. Adeyemo, J.A., Otieno, F.A.O.: Optimizing planting areas using differential evolution (DE) and linear programming (LP). *Journal of Physical Sciences* 4(4), 212–220 (2009)
14. Adeyemo, J.A., Otieno, F.A.O.: Multi-objective differential evolution algorithm (MDEA) for solving engineering problems. *Journal of Applied Sciences* 9(20), 3652–3661 (2009)
15. Babel, M.S., Gupta, A.D., Nayak, D.K.: A Model for Optimal Allocation of Water to Competing Demands. *Journal of Water Resources Management* 19, 693–712 (2005)
16. Fernandes, M., Schreider, S.: A penalty minimisation resource (water) allocation model to simulate the effects of new infrastructure in the Goulburn irrigation system. In: 18th World IMACS / MODSIM Congress, Cairns, Australia, July 13–17 (2009)
17. Otieno, F.A.O., Adeyemo, J.A.: Strategies of differential evolution for optimum cropping pattern. *Trends in Applied Sciences Research* 5, 1–15 (2010)
18. Cai, X., McKinney, D.C., Lasdon, L.S.: Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. *Advances in Water Resources* 24(6), 667–676 (2001)

19. Yuan, X., Zhang, Y., Wang, L., Yuan, Y.: An enhanced differential evolution algorithm for daily optimal hydro generation scheduling. *Computers & Mathematics with Applications* 55(11), 2458–2468 (2008)
20. Selle, B., Muttil, N.: Testing the structure of a hydrological model using Genetic Programming. *Journal of Hydrology* 397, 1–9 (2010)
21. Weise, T.: *Global Optimization Algorithms - Theory and Application*, vol. 1., p. 820 (2009), Thomas Weise: <http://www.it-weise.de/projects/book.pdf>
22. Karterakis, S.M., Karatzas, G.P., Nikolos, L.K., Papadopoulou, M.P.: Application of Linear Programming and Differential Evolutionary Optimization Methodologies for the Solution of Coastal Subsurface Water Management Problems Subject to Environmental Criteria. *Journal of Hydrology* 342, 270–282 (2007)
23. Qin, H., Zhou, J., Lu, Y., Wang, Y., Zhang, Y.: Multi-objective differential evolution with adaptive Cauchy mutation for short-term multi-objective optimal hydrothermal scheduling. *Energy Conversion and Management* 51(4), 788–794 (2010)
24. Yousefi, H., Handroos, H., Soleymani, A.: Application of Differential Evolution in system identification of a servo-hydraulic system with a flexible load. *Mechatronics* 18, 513–528 (2008)
25. Nasserri, M., Asghari, K., Abedini, M.J.: Optimized Scenario for Rainfall Forecasting Using Genetic Algorithm Coupled with Artificial Neural Network. *Expert Systems with Applications* 35, 1415–1421 (2008)
26. Cho, J.H., Seok Sung, K., Ryong Ha, S.: A river water quality management model for optimising regional wastewater treatment using a genetic algorithm. *Journal of Environmental Management* 73, 229–242 (2004)
27. Kerachian, R., Karamouz, M.: A stochastic conflict resolution model for water quality management in reservoir-river systems. *Advances in Water Resources* 30, 866–882 (2007)
28. Wang, J.Y., Chang, T.P., Chen, J.S.: An enhanced genetic algorithm for bi-objective pump scheduling in water supply. *Expert Systems with Applications* 36, 10249–10258 (2009)
29. Wang, Y., Wang, H., Lei, X., Jiang, Y., Song, X.: Flood Simulation using Parallel Genetic Algorithm Integrated Wavelet Neural Networks. *Journal of Neurocomputing* 74(17), 2734–2744 (2011)
30. Lavric, V., Iancu, P., Pleay, V., Barbosa-PaVoa, A., Matos, H.: Optimal water system topology through genetic algorithm under multiple contaminated-water sources constraint. In: *Computer Aided Chemical Engineering: European Symposium on Computer-Aided Process Engineering-14*, vol. 18, pp. 433–438 (2004)
31. Lavric, V., Iancu, P., Pleay, V.: Genetic algorithm optimisation of water consumption and wastewater network topology. *Journal of Cleaner Production* 13, 1405–1415 (2005)
32. Price, K., Storn, R.M., Lampinen, J.A.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag New York, Inc., Secaucus (2005)
33. Goudos, S.K., Baltzis, K.B., Antoniadis, K., Zaharis, Z.D., Hilas, C.S.: A comparative study of common and self-adaptive differential evolution strategies on numerical benchmark problems. *Procedia Computer Science* 3, 83–88 (2011)
34. Lu, Y., Zhou, J., Qin, H., Wang, Y., Zhang, Y.: Environmental/economic dispatch problem of power system by using an enhanced multi-objective differential evolution algorithm. *Energy Conversion and Management* 52, 1175–1183 (2011)
35. Price, K., Storn, R.: Differential evolution. *Dr. Dobb's Journal*, 18–24 (1997)
36. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001)

37. Karterakis, S.M., Karatzas, G.P., Nikolos, L.K., Papadopoulou, M.P.: Application of linear programming and differential evolutionary optimization methodologies for the solution of coastal subsurface water management problems subject to environmental criteria. *Journal of Hydrology* 342, 270–282 (2007)
38. Mirghani, B.Y., Mahinthakumar, K.G., Tryby, M.E., Ranjithan, R.S., Zechman, E.M.: A parallel evolutionary strategy based simulation optimization approach for solving groundwater source identification problems. *Advances in Water Resources* 32, 1373–1385 (2009)
39. Berlich, R.D., Kunze, M.: Parametric optimization with evolutionary strategies in particle physics: Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. In: IXth International Workshop on Advanced Computing and Analysis Techniques in Physics Research
40. Kanoun, O., Troltzsch, U., Trankler, H.R.: Benefits of evolutionary strategy in modeling of impedance spectra. *Electrochimica Acta* 51, 1453–1461 (2006)
41. Navale, R.L., Nelson, R.M.: Use of evolutionary strategies to develop an adaptive fuzzy logic controller for a cooling coil. *Energy and Buildings* 42, 2213–2218 (2010)
42. Aytok, A., Asce, M., Alp, M.: An Application of Artificial Intelligence for Rainfall-Runoff Modeling. *Journal of Earth System Science* 117(2), 145–155 (2008)
43. Ghorbani, M.A., Khatibi, R., Aytok, A., Makarynsky, O., Shiri, J.: Sea water level forecasting using genetic programming and comparing the performance with Artificial Neural Networks. *Computers & Geosciences* 36, 620–627 (2010)
44. Nasser, M., Moeini, A., Tabesh, M.: Forecasting monthly urban water demand using Extended Kalman Filter and Genetic Programming. *Expert Systems with Applications* 38, 7387–7395 (2011)
45. Shiri, J., Kisi, O.: Comparison of genetic programming with neuro-fuzzy systems for predicting short-term water table depth fluctuations. *Computers & Geosciences* 37, 1692–1710 (2010)
46. Kisi, O., Guven, A.: A machine code-based genetic programming for suspended sediment concentration estimation. *Advances in Engineering Software* 41, 939–945 (2010)
47. Sreekanth, J., Datta, B.: Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models. *Journal of Hydrology* 393, 245–256 (2010)
48. Kerkez, B., Glaser, S.D., Dracup, J.A., Bales, R.C.: A Hybrid System Model of Seasonal Snowpack Water Balance. In: HSCC 2010, April 12–15 (2010)
49. Chen, S., Fu, G.: Combining fuzzy iteration model with dynamic programming to solve multiobjective multistage decision making problems. *Fuzzy Sets and Systems* 152, 499–512 (2005)
50. Khademi, M.H., Setoodeh, P., Rahimpour, M.R., Jahanmiri, A.: Optimization of methanol synthesis and cyclohexane dehydrogenation in a thermally coupled reactor using differential evolution (DE) method. *International Journal of Hydrogen Energy* 34, 6930–6944 (2009)
51. Cisty, M.: Hybrid model for water distribution design. *Congress on Evolutionary Computation (CEC)*, 1–8 (2010)

# Author Index

- Adeyemo, Josiah, [490](#)  
Alvarado, Sergio, [151](#)  
Alvarado-Velazco, Paola Berenice, [385](#)  
Avigad, Gideon, [120](#) [135](#) [151](#) [395](#)  
Avilés, Carlos, [284](#)  
Ayala-Ramirez, Victor, [385](#)
- Blanchet, Jose, [18](#)  
Botello, Salvador, [220](#)  
Bouvry, Pascal, [206](#)
- Castro, Carlos, [170](#)  
Charan Kumari, A., [105](#)  
Chen, Stephen, [414](#)  
Clemente, Eddie, [337](#) [353](#) [370](#)  
Coello Coello, Carlos A., [151](#) [265](#)  
Coria, Luis, [56](#)  
Crawford, Broderick, [170](#)  
Cuevas, Erik, [297](#) [311](#)  
Cuevas, Francisco, [325](#)
- Diaz-Diaz, Elva, [235](#) [446](#)  
Dominguez-Medina, Christian, [88](#)  
Dozal, León, [337](#) [353](#) [370](#)  
Dunn, Enrique, [70](#)
- Eisenstadt, Erella, [120](#) [395](#)
- Fournier, René, [414](#)
- Gómez-Meneses, Pedro, [430](#)  
Gadelha Guimar, Frederico, [395](#)  
Galvan, Edgar, [40](#)  
Garro, Beatriz A., [284](#)
- Glizer, Valery Y., [120](#)  
Glynn, Peter, [18](#)  
Goldvard, Alex, [135](#)  
Guijarro, Enrique, [56](#)  
Gupta, M. P., [105](#)  
Gutierrez-Marfileno, Luis Fernando, [446](#)
- Hernández, Daniel, [337](#)  
Hernández-Aguirre, Arturo, [220](#)  
Hettenhausen, Jan, [414](#)
- Ibarra-Martinez, Leonice, [446](#)
- Jiménez, Julio, [325](#)
- Kattan, Ahmed, [40](#)  
Khadraoui, Djamel, [206](#)  
Kipouros, Timoleon, [474](#)
- Lara, Adriana, [151](#)  
Lewis, Andrew, [414](#) [430](#)
- Mansour, Nashat, [461](#)  
Martínez, Yuliana, [56](#)  
McDermott, James, [40](#)  
Monfroy, Eric, [170](#)  
Muñoz Zavala, Angel Eduardo, [249](#)
- Naredo, Enrique, [70](#)
- Olague, Gustavo, [284](#) [337](#) [353](#) [370](#)  
Olofintoye, Oluwatosin, [490](#)  
Osuna, Valentín, [297](#) [311](#)  
Otieno, Fred, [490](#)

- Pérez-Cisneros, Marco, [311](#)  
Palma, Wenceslao, [170](#)  
Paredes, Fernando, [170](#)  
Ponce-de-Leon, Eunice, [235](#), [446](#)
- Röning, Juha, [206](#)  
Randall, Marcus, [414](#), [430](#)  
Richer, Jean-Michel, [188](#)  
Rodriguez-Tello, Eduardo, [188](#)  
Rudolph, Günter, [88](#)
- Salomon, Shaul, [135](#), [151](#), [395](#)  
Schütze, Oliver, [88](#), [135](#), [151](#)  
Schaberreiter, Thomas, [206](#)  
Sossa, Humberto, [284](#), [297](#), [311](#), [325](#)  
Sotelo, Arturo, [56](#)
- Soto, Ricardo, [170](#)  
Srinivas, K., [105](#)  
Sun, Jian-Qiao, [3](#)
- Tarhini, Abbas, [461](#)  
Trautmann, Heike, [88](#)  
Trujillo, Leonardo, [40](#), [56](#), [70](#)
- Valdez, Ivvan, [220](#)  
Vazquez-Ortiz, Karla E., [188](#)  
Villegas, Juan, [284](#)  
Villela Tinoco, José Carlos, [265](#)
- Zaldivar, Daniel, [311](#)  
Zeitunlian, Hratch, [461](#)  
Zheng, Shuheng, [18](#)