Advances in Intelligent Systems and Computing 288

Alexandru-Adrian Tantar · Emilia Tantar
Jian-Qiao Sun · Wei Zhang
Qian Ding · Oliver Schütze
Michael Emmerich · Pierrick Legrand
Pierre Del Moral · Carlos A. Coello Coello *Editors*

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

Springer

Advances in Intelligent Systems and Computing

Volume 288

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

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

Alexandru-Adrian Tantar · Emilia Tantar
Jian-Qiao Sun · Wei Zhang
Qian Ding · Oliver Schütze
Michael Emmerich · Pierrick Legrand
Pierre Del Moral · Carlos A. Coello Coello
Editors

EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V

Springer

Editors

Alexandru-Adrian Tantar
Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg
Luxembourg

Emilia Tantar
Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg
Luxembourg

Jian-Qiao Sun
University of California
Merced California
USA

Wei Zhang
Beijing University of Technology
College of Mechanical Engineering
Beijing, China

Qian Ding
Department of Mechanics
Tianjin University
Tianjin, China

Oliver Schütze
CINVESTAV-IPN, Mexico
Mexico

Michael Emmerich
Leiden Institute of Advanced Computer Science
Leiden University
The Netherlands

Pierrick Legrand
Université de Bordeaux
Bordeaux
France

Pierre Del Moral
School of Mathematics and Statistics
University of New South Wales
New South Wales
Australia

Carlos A. Coello Coello
CINVESTAV-IPN, Mexico
Mexico

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-07493-1 ISBN 978-3-319-07494-8 (eBook)
DOI 10.1007/978-3-319-07494-8
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014939944

Printed on acid-free paper

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

Preface

This volume encloses a series of research articles and contributions that were presented at the EVOLVE 2014 International Conference in Beijing, China, July 1–4, 2014.

The aim of the EVOLVE (http://www.evolve-conference.org) conference series is to build a bridge between probability, statistics, set oriented numerics and evolutionary computing, as to identify new common and challenging research aspects. The event is intended to foster a growing interest for robust and efficient new methods with a sound theoretical background. Furthermore, EVOLVE aspires at unifying theory and applied cutting-edge techniques that ensure performance guarantee factors. The massive use and large applicability spectrum of evolutionary algorithms in real-life applications, as an example, determined a need for establishing solid theoretical grounds. In a similar way, one may consider mathematical objects that are sometimes difficult and/or costly to calculate, namely in the light of acknowledged new results which show that evolutionary algorithms can act in some cases as good and fast estimators. 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. And examples could continue. What this collection is in the end about and what common practice confirms in many cases is that theory-based and applied results have to be considered in a unified perspective. The volume is thus focused on challenging aspects arising at the passage from theory to new paradigms and practice, aiming to provide a unified view while, at the same time, raising questions related to reliability, performance guarantees and modeling. As a consequence of these aims and by gathering researchers with different backgrounds, e.g. computer science, mathematics, statistics and physics, a unified view and vocabulary can emerge where theoretical advancements may echo throughout different domains.

The volume gathers contributions that emerged from the conference tracks, ranging from probability to set oriented numerics and evolutionary computation; all complemented by the bridging purpose of the conference, e.g. *Complex Networks and Landscape Analysis*, or by the more application oriented perspective. The novelty of the volume, when considering the EVOLVE series, comes from targeting also the practitioner's view. This is supported by the *Machine Learning Applied to Networks* and *Practical Aspects of Evolutionary Algorithms* tracks, providing surveys on new (some of them unhandled) application areas, as in the networking area and useful insights in the development of evolutionary techniques, from a practitioner's perspective. Complementary to these directions, the conference tracks supporting the volume, follow on the individual advancements of the subareas constituting the scope of the conference, through the *Computational Game Theory*, *Local Search and Optimization*, *Genetic Programming*, *Evolutionary Multi-objective optimization* tracks.

As an ending thought, our gratitude goes out to all the invited speakers for accepting to give an outstanding presentation and an overview of their latest work during the event, to all the authors, for sharing their knowledge and expertise and, last but not least, to all the participants for their extraordinary support. We would also like to express our foremost appreciation to all the referees and members of the program committee which, through their considerate work, contributed to the creation of a bridge between the different fields covered by the event. The editors would furthermore like to thank Dr. Thomas Ditzinger and Professor Janusz Kacprzyck (Editor-In-Chief, Springer Studies in Computational Intelligence Series) for the extraordinary collboration, since the begining of the EVOLVE series and this during the entire editing process. Finally, we would like to gratefully thank the partner institutions and the sponsors of the event which all made EVOLVE 2014 possible and a great success.

Luxembourg, Merced, Beijing, Mexico City
Leiden, Bordeaux, Sydney

March 2014
http://www.evolve-conference.org

Alexandru-Adrian Tantar
Emilia Tantar
Jian-Qiao Sun
Wei Zhang
Qian Ding
Oliver Schütze
Michael Emmerich
Pierrick Legrand
Pierre Del Moral
Carlos A. Coello Coello

Organization

EVOLVE 2014 was jointly organized by the University of California, Merced, USA, the Tianjin University (Peiyang University), China, CINVESTAV-IPN (Research and Advanced Studies Center of the National Polytechnic Institute of Mexico), Mexico, the University of Luxembourg, Luxembourg, and INRIA (National Institute for Research in Computer Science and Control), France. The Beijing University of Technology, China, and CONACYT, Mexico, were also among the partners of the event. Additional details about the event can be found online at http://www.evolve-conference.org.

EVOLVE Series Chairs

Pierre Del Moral	University of New South Wales, Australia
Michael Emmerich	Leiden University, The Netherlands
Pierrick Legrand	University of Bordeaux 2, France
Oliver Schütze	CINVESTAV-IPN, Mexico
Alexandru-Adrian Tantar	University of Luxembourg, Luxembourg
Emilia Tantar	University of Luxembourg, Luxembourg

EVOLVE 2014 Organizing Committee

General Chair

Haiyan Hu	Beijing Institute of Technology, China

Local Chairs

Jian-Qiao Sun	University of California, Merced, USA
Wei Zhang	Beijing University of Technology, China
Qian Ding	Tianjin University, China

Proceedings Chair

Alexandru-Adrian Tantar	University of Luxembourg, Luxembourg

Students Chair

Pierrick Legrand	University of Bordeaux 2, France

Finance Chair

Oliver Schütze	CINVESTAV-IPN, Mexico
Xiang-Ying Guo	Beijing University of Technology, China

Tutorials Chair

Emilia Tantar	University of Luxembourg, Luxembourg

Website

Alexandru-Adrian Tantar	University of Luxembourg, Luxembourg
Emilia Tantar	University of Luxembourg, Luxembourg

Secretary

Xiang-Ying Guo	Beijing University of Technology, China
Xiao-Dong Yang	Beijing University of Technology, China

EVOLVE 2014 Session Chairs

Michael T. M. Emmerich
Complex Networks and Landscape Analysis

Jian-Qiao Sun
Multi-Objective Optimal Design and Control of Nonlinear Systems with Cell Mapping Methods

Jörn Mehnen
Practical Aspects of Evolutionary Algorithms

Kukliński Sławomir, Emilia Tantar

Machine Learning Applied to Networks

Oliver Schütze
Set Oriented Numerics

Leonardo Trujillo, Edgar Galvan-Lopez and Pierrick Legrand
Genetic Programming

Heike Trautmann, Günter Rudolph, Oliver Schütze
Evolutionary Multiobjective Optimization

Rodica Ioana Lung, Dan Dumitrescu
Computational Game Theory

Massimiliano Vasile
Robust Optimization

Program Committee

All full papers were peer reviewed. We thank the referees for their voluntary effort.

J.A. Gonzalez
J. Adeyemo
V. Basto-Fernandes
E. Chavez
F. Chicano
E. Clemente
C. Cotta
N. Cruz Cortés
A. Deutz
A. Duda
D. Dumitrescu
M. Emmerich
J. Fernandez
F. Fernandez-De-Vega
A. Galis
E. Galvan-Lopez
C. Grimme
C. Hernandez
R.I. Lung
A. Kattan
T. Kipouros
S. Kukliński
M. López-Ibáñez
R. Landa
A. Lara
P. Legrand
R. Li
Y. Li
J. Liu
G. Luque
L. Martí
J. Mehnen
A. Oulamara
R. Preis
R. Purshouse
E. Rodríguez Tello
G. Rudolph
H. Sato
C. Schommer
O. Schuetze
C. Segura
S. Silva
J.Q. Sun
A.A. Tantar
E. Tantar
H.N. Teodorescu
H. Trautmann
L. Trujillo
M. Vasile
T. Wagner
W. Wang
S. Wessing
I. Yevseyeva
A. Zomaya

EVOLVE 2014 Partner Institutions

University of California, Merced, USA
Tianjin University (Peiyang University), China
Beijing University of Technology, China
CINVESTAV-IPN, Mexico
CONACYT, Mexico
University of Bordeaux 2, France
University of Luxembourg, Luxembourg

EVOLVE Series Website: http://www.evolve-conference.org

Contents

Part I: Set Oriented Numerics

Part II: Computational Game Theory

Part III: Machine Learning Applied to Networks

Part IV: Complex Networks and Landscape Analysis

Part V: Local Search and Optimization

Part VI: Genetic Programming

Part VII: Evolutionary Multiobjective Optimization

Part VIII: Practical Aspects of Evolutionary Algorithms

Part I

Set Oriented Numerics

Response Analysis of a Forced Duffing Oscillator with Fuzzy Uncertainty

Ling Hong[1,⋆], Jun Jiang[1], and Jian-Qiao Sun[2]

[1] State Key Lab for Strength and Vibration
Xi'an Jiaotong University
Xi'an 710049, China
hongling@mail.xjtu.edu.cn
[2] School of Engineering
University of California at Merced
Merced, CA 95344, USA

Abstract. The transient and steady-state membership distribution functions (MDFs) of fuzzy response of a forced Duffing oscillator with fuzzy uncertainty are studied by means of the Fuzzy Generalized Cell Mapping (FGCM) method. The FGCM method is first introduced. A rigorous mathematical foundation of the FGCM is established with a discrete representation of the fuzzy master equation for the possibility transition of continuous fuzzy processes. The FGCM offers a very effective approach for solutions to the fuzzy master equation based on the min-max operator of fuzzy logic. Fuzzy response is characterized by its topology in the state space and its possibility measure of MDFs. The response topology is obtained based on the qualitative analysis of the FGCM involving the Boolean operation of 0 and 1. The evolutionary process of transient and steady-state MDFs is determined by the quantitative analysis of the FGCM with the min-max calculations. It is found that the evolutionary orientation of MDFs is in accordance with invariant manifolds leading to invariant sets. In the evolutionary process of a steady-state fuzzy response with an increase of the intensity of fuzzy noise, a merging bifurcation is observed in a sudden change of the MDFs from two sharp peaks of most possibility to one peak band around unstable manifolds.

Keywords: Fuzzy Uncertainty, Possibility Measure, Fuzzy Response, Membership Distribution Function, Generalized Cell Mapping.

1 Introduction

Engineering systems are often subjected to uncertainties that are associated with the lack of precise knowledge of system parameters and operating conditions and that are originated from variability in manufacturing processes. The uncertainties can have significant influence on the dynamic response and the reliability of the system, and are often modeled as random variables or fuzzy sets. This paper proposes the FGCM method to analyze the response of nonlinear dynamical systems with fuzzy uncertainties. Specifically, we are interested in a nonlinear dynamical system whose response

[⋆] Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_1, © Springer International Publishing Switzerland 2014

is a fuzzy process, and study the transient and steady-state membership distribution functions (MDFs) of fuzzy response.

For fuzzy nonlinear dynamical systems, a response process is difficult to analyze because the evolution of the MDFs of the fuzzy response process cannot be readily obtained analytically. Many studies dealt with fuzzy dynamical systems governed by linear ordinary differential equations [1–3]. Chaotic sequences of fuzzy nonlinear maps were studied [4]. A master equation was derived for the evolution of MDFs of fuzzy processes [5,6]. However, the solution to the fuzzy master equation is rare, particularly for nonlinear dynamical systems. The theory on the evolution of the MDFs of fuzzy processes is far less complete and mature compared to that for the probability density of stochastic processes [7]. The first attempt to describe the dynamics of fuzzy systems was done in 1973 by Nazaroff [8], who investigated fuzzy topological polysystems. From that time several approaches to describing fuzzy dynamics were proposed. For instance, Kloeden [9] defined and considered the extension of dynamical systems to fuzzy states. However, the notion of fuzzy velocity was not introduced there and an explicit differential equation for continuous evolution was not given. Another approach to fuzzy dynamics using differential inclusions was developed by Aubin and others [10]. The notion of fuzzy set of velocities was introduced and the standard equations of evolution were fuzzyfied. Since these equations were obtained outside fuzzy logic framework, the fuzzy differential inclusion leads to the problem that the restriction of the values of the MDFs to the interval [0,1] was not preserved by the evolution in general. For the same reason, similar problems occur in the solution of the Cauchy problem for fuzzy differential equation [11].

Fuzzy response is naturally global in the sense that it is represented by a fuzzy set of a finite possibility measure in the state space. It is computationally intensive and ineffective to study such a solution by using numerical simulations [12, 13]. The cell mapping method represents a major advancement in this regard. Chen and Tsao pioneered the work of fuzzy control design with the simple cell mapping method [14,15]. Smith and his associates further extended the simple cell mapping method to fuzzy optimal control problems in higher dimensional state space [16, 17]. The generalized cell mapping (GCM) method offers a probabilistic tool for the global analysis of nonlinear dynamical systems [18]. Edwards and Choi proposed to impose the initial probability distribution in each cell that has the same form as a typical fuzzy membership function [19]. However, they did not apply Zadeh's extension principle to manipulate fuzzy sets. The resulting generalized cell mapping, however, is still a Markov chain. Sun and Hsu extended the generalized cell mapping to fuzzy systems by applying Zadeh's extension principle [20]. A variety of random vibration and stochastic optimal problems were studied by Sun, Hsu and their associates [21,22]. Crises in chaotic dynamical systems were investigated by Hong and Xu using the GCM with digraphs [23, 24]. More recently, the GCM method was applied to study bifurcations of fuzzy nonlinear dynamical systems [25,26].

The remainder of the paper is outlined as follows. In Section 2, we introduce the FGCM method, and discuss its properties. In Section 3, we study the evolution of transient and steady-state MDFs of a forced Duffing oscillator with fuzzy noise. The paper concludes in Section 4.

2 Fuzzy Generalized Cell Mapping

2.1 Discrete Representation of the Fuzzy Master Equation

A rigorous mathematical foundation of the FGCM method can be established. Consider the fuzzy master equation for the possibility transition of continuous fuzzy processes [3,5,6],

$$p(\mathbf{x},t) = \sup_{\mathbf{x}_0 \in \mathbf{D}} \left[\min\{p(\mathbf{x},t|\mathbf{x}_0,t_0), p(\mathbf{x}_0,t_0)\} \right], \quad \mathbf{x} \in \mathbf{D} \tag{1}$$

where $\mathbf{x}$ is a fuzzy process, $p(\mathbf{x},t)$ is the membership distribution function of $\mathbf{x}$ at t, and $p(\mathbf{x},t|\mathbf{x}_0,t_0)$ is the transition possibility function, also known as a fuzzy relation [3]. $\mathbf{D}$ is a bounded domain of interest in the state space. A partial differential equation from Equation (1) for continuous time processes has been derived by Friedman and Sandler [5, 6]. This equation is analogous to the Fokker-Planck-Kolmogorov equation for the probability density function of stochastic processes [7]. The solution to this equation is in general very difficult to obtain analytically.

Next we show that FGCM can be introduced as a discrete representation of the fuzzy master equation (1). Consider a dynamical system with fuzzy uncertainty.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},t,S), \quad \mathbf{x} \in \mathbf{D}, \tag{2}$$

where $\mathbf{x}$ is the state vector, t the time variable, S a fuzzy set with a membership function $\mu_S(s) \in (0,1]$ where $s \in S$, and $\mathbf{f}$ is a vector-valued nonlinear function of its arguments. When the system parameter S is a fuzzy number. Equation (2) is a fuzzy differential equation.

The cell mapping method proposes to further discretize the time and state variables in searching for the global solution of the system [18]. In order to apply the cell mapping method, we also need to discretize the fuzzy set S. We divide S into M segments of appropriate length and sample a value $s_k \in S$ $(k = 1, \cdots, M)$ in the middle of each segment. The division of S is such that there is at least one s_k with membership grade equal to one.

The domain $\mathbf{D}$ is then discretized into N small cells. Each cell is identified by an integer ranging from 1 to N. For a cell, say cell j, N_p points are uniformly sampled from cell j. By applying the method of numerical simulation, we generate $M \times N_p$ fuzzy sample trajectories of one period T long. The length T is taken to be one mapping step. Each trajectory carries a membership grade determined by that of s_k's. We then find the cells in which the end points of the trajectories fall. Assume that cell i is one of the image cells of cell j, and that there are m $(0 < m \leqslant MN_p)$ trajectories falling in cell i. Define a quantity

$$p_{ij} = \max_{i_k} [\mu_S(s_{i_k})], \quad 0 < p_{ij} \leqslant 1, \tag{3}$$

where i_k $(k = 1, 2, ..., m)$ are referred to the trajectories falling in cell i, and $\mu_S(s_{i_k})$ are the membership grades of the corresponding trajectories. This procedure for computing p_{ij} is known as the sampling point method in the context of generalized cell mapping [18]. We should note that partition of the domain $\mathbf{D}$ and fuzzy set S need not be uniform.

Now, assume that the membership grade of the system being in cell j at the n^{th} mapping step is $p_j(n)$ $(0 < p_j(n) \leqslant 1)$. Cell j is mapped in one step to cell i with the membership grade given by

$$\max\left\{\min\left[\mu_S\left(s_{i_1}\right), p_j\left(n\right)\right], \min\left[\mu_S\left(s_{i_2}\right), p_j\left(n\right)\right], \cdots, \min\left[\mu_S\left(s_{i_m}\right), p_j\left(n\right)\right]\right\} \tag{4}$$

$$= \min[\max_{i_k}\left(\mu_S\left(s_{i_k}\right)\right), p_j\left(n\right)] = \min[p_{ij}, p_j\left(n\right)].$$

Considering all possible pre-images of cell i, we have the membership grade of the system being in cell i at the $(n+1)^{th}$ step as

$$p_i\left(n+1\right) = \max_j \min\left[p_{ij}, p_j\left(n\right)\right]. \tag{5}$$

Let $\mathbf{p}\left(n\right)$ be a vector with components $p_i\left(n\right)$, and $\mathbf{P}$ a matrix with components p_{ij}. Equation (5) can be written in a compact matrix notation

$$\mathbf{p}\left(n+1\right) = \mathbf{P} \circ \mathbf{p}\left(n\right), \ \ \mathbf{p}\left(n\right) = \mathbf{P}^n \circ \mathbf{p}\left(0\right), \tag{6}$$

where $\mathbf{P}^{n+1} = \mathbf{P} \circ \mathbf{P}^n$ and $\mathbf{P}^0 = \mathbf{I}$. $\circ$ denotes the min-max operation. The matrix $\mathbf{P}$ denotes the one-step transition membership matrix. $\mathbf{P}^n$ denotes the n-step transition possibility matrix. The vector $\mathbf{p}\left(n\right)$ is called the n-step membership distribution vector, and $\mathbf{p}\left(0\right)$ the initial membership distribution vector. The $(i,j)^{th}$ element p_{ij} of the matrix $\mathbf{P}$ is called the one-step transition membership from cell j to cell i.

Equation (6) is called a fuzzy generalized cell mapping (FGCM) system, which describes the evolution of the fuzzy solution process $\mathbf{x}\left(t\right)$ and its MDFs $p(\mathbf{x}, t)$, and is a finite approximation to the fuzzy dynamical system (2) in $\mathbf{D}$.

Equation (5) of the FGCM can be viewed as a discrete representation of Equation (1). The FGCM offers a very effective method for solutions to this equation, particularly, for fuzzy nonlinear dynamical systems.

2.2 The Qualitative and Quantitative Properties of FGCM

Qualitative Properties. Recall that the min-max operation in Eq. (5) really represents the intersection (product) and union (summation) of fuzzy sets in the form of cells in $\mathbf{D}$. Hence, $\mathbf{P} \circ \mathbf{p}$ is an inner product of fuzzy sets. The topological matrix of $\mathbf{P}$, denoted by $[\overline{p}_{ij}]$, and the topological vector of $\mathbf{p}\left(n\right)$, denoted by $\{\overline{p}_i(n)\}$, are defined as

$$\overline{p}_{ij} = \begin{cases} 1, p_{ij} > 0 \\ 0, p_{ij} = 0 \end{cases}, \ \overline{p}_i(n) = \begin{cases} 1, p_i(n) > 0 \\ 0, p_i(n) = 0 \end{cases}, \ n \geq 0. \tag{7}$$

Topologically, Eq. (6) becomes

$$\overline{\mathbf{p}}\left(n+1\right) = \overline{\mathbf{P}} \circ \overline{\mathbf{p}}\left(n\right), \ \ \overline{\mathbf{p}}\left(n\right) = \overline{\mathbf{P}}^n \circ \overline{\mathbf{p}}\left(0\right), \tag{8}$$

where $\overline{\mathbf{P}}^{n+1} = \overline{\mathbf{P}} \circ \overline{\mathbf{P}}^{n}$ and $\overline{\mathbf{P}}^{0} = \mathbf{I}$. Note that the min-max operation is equivalent to the logic operations of multiplication $\wedge$ and addition $\vee$ of binary numbers: $0 \wedge 1 = 0$, $1 \wedge 0 = 0, 0 \wedge 0 = 0, 1 \wedge 1 = 1, 0 \vee 1 = 1, 1 \vee 0 = 1, 0 \vee 0 = 0$, and $1 \vee 1 = 1$. Hence, the min-max operation in Eq. (8) leads to the identical result to that of the topological matrix of the Markov chains. Hence, $\overline{\mathbf{P}}$ forms a digraph and can be partitioned to identify persistent groups of cells representing stable solutions and the transient cells including the unstable solutions [27].

Quantitative Properties. Under the min-max operation in Eq. (5), the membership function of the cells is non-increasing in the sense that,

$$||\mathbf{p}\,(n+1)\,|| \leq ||\mathbf{p}\,(n)\,||,\ \forall n \geq 0, \tag{9}$$

where the norm of the fuzzy membership vector is defined as

$$||\mathbf{p}\,(n)\,|| = \max_{i}\{p_i\,(n)\}. \tag{10}$$

Furthermore, the min-max operator does not introduce any new numbers that are not the entries of the matrix $\mathbf{P}$ or the vector $\mathbf{p}(0)$. This implies that $\mathbf{p}(n)$ can assume only finite number of possible values as $n \to \infty$. In the steady state, $\mathbf{p}(n)$ will either converge to a constant vector, or to a set of vectors which form a periodic group and repeat themselves as the iteration goes on. In either case, we consider the system to have converged to the steady state. Because there are only finite number of possible values for $\mathbf{p}(n)$, Eq. (5) will converge in finite number of iterations. This is a sharp contrast to Markov chains, which theoretically converge to steady state in infinite iterations.

The qualitative and quantitative properties lead to a dichotomy in the computation treatment of the FGCM, and ensure the accuracy and efficiency of the FGCM. Boolean operations are only used in the qualitative analysis of the FGCM, while the min-max operations are only involved in the quantitative analysis of FGCM. As a result, the transient and steady-state MDFs of a fuzzy response process can be effectively determined in a new way.

3 Response Analysis of a Forced Duffing Oscillator with Fuzzy Uncertainty

3.1 The Global Properties of Deterministic Forced Duffing Oscillator

Consider the forced Duffing oscillator.

$$\ddot{x} + \kappa\dot{x} + x^3 = B\cos t. \tag{11}$$

It is a mathematical model of various physical systems where the damping κ and forcing magnitude B are system parameters. The system has various types of steady state motions depending on system parameters (κ, B) as well as initial conditions. In the present work, we are interested in the region of periodic motions. In particular, we take (κ, B)=(0.2, 0.3) where the system has two coexistent period-one attractors $\mathbf{A}_1$ and $\mathbf{A}_2$.

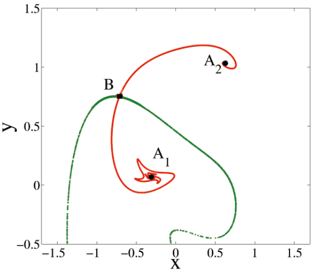


Fig. 1. Global phase portrait of the deterministic Duffing (11) with κ=0.2 and B=0.3. A$_1$ and A$_2$ denote two period-one attractors. On basin boundary lies the saddle denoted by B. The basin boundary denoted by a green line is the stable manifold of the saddle B. The unstable manifolds of the saddle B denoted by a red line are directed to A$_1$ and A$_2$ respectively.

In Fig. 1 is shown the global properties of the equation (11) by means of point mapping under cell reference (PMUCR), a two-scaled numerical global analysis method [28,29]. The basin boundary (a green line) is the stable manifolds of the unstable saddle B. The two branches (a red line) of the unstable manifolds of B are directed to the two fixed points A$_1$ and A$_2$ respectively. B is the unstable invariant set of the Duffing system (11). A$_1$ and A$_2$ the stable invariant sets.

In the following sections, we will study transient and steady-state MDFs of a fuzzy response for the forced Duffing oscillator with fuzzy uncertainty.

3.2 Fuzzy Response

Consider now the Duffing equation with fuzzy noise

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -x^3 - 0.2y + S\cos t. \end{aligned} \tag{12}$$

where S is a fuzzy parameter of the forcing amplitude with a triangular membership function,

$$\mu_S(s) = \begin{cases} [s-(s_0-\varepsilon)]/\varepsilon, & s_0-\varepsilon \leqslant s < s_0 \\ -[s-(s_0+\varepsilon)]/\varepsilon, & s_0 \leqslant s < s_0+\varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$\varepsilon > 0$ is a parameter characterizing the intensity of fuzziness of S and is called a fuzzy noise intensity. s_0 is the nominal value of S with membership grade $\mu_S(s_0) = 1$. In the computation, we take $s_0 = 0.301$.

The domain $\mathbf{D} = (-1.5 \leq x \leq 1.7) \times (-0.5 \leq y \leq 1.5)$ is discretized into 141×141 cells. The 5×5 sampling points are used within each cell. S is discretized into 401 segments. Hence, out of each cell, there are 10,025 trajectories with varying membership grades to determine the one-step transition possibility with the time length $\triangle T = T = 2\pi$. T is called one mapping step. The FGCM is used to analysis the transient and steady-state MDFs of the fuzzy response of the system (12).

Transient Analysis of MDFs. In this section, we take the intensity of fuzzy noise $\varepsilon = 0.07$. Figs. 2, 3, and 4 show the evolutionary process of transient MDFs of displacement x and velocity y with initial possibility distributions $p_{n_1}(0) = 1$ and $p_{n_2}(0) = 1$. n_1 and n_2 denote the persistent groups representing the two attractors A_1 and A_2.. In Fig. 2 is shown the time evolution of marginal MDFs for x and y respectively. In Figs. 3 and 4 are respectively shown the contour and 3D surface plots of joint MDFs of x and y. The evolutionary trend to follow in Figs. 3 and 4 is that as the time increases we move from (a) to (b) in (c) and (d), two peaks of possibility one of MDFs move respectively along the two branches of the unstable manifold of B and converge around A_1 and A_2.

Steady-State Analysis of MDFs. We study the evolutionary process of the steady-state fuzzy response of the noisy Duffing system (12) as the intensity of fuzzy noise

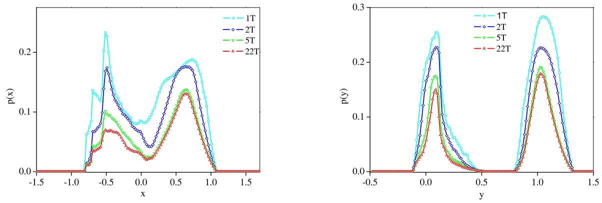


Fig. 2. Transient marginal MDFs of displacement x and velocity y for the fuzzy Duffing equation (12) with initial possibility distributions $p_{n_1}(0)$=1 and $p_{n_2}(0)$=1

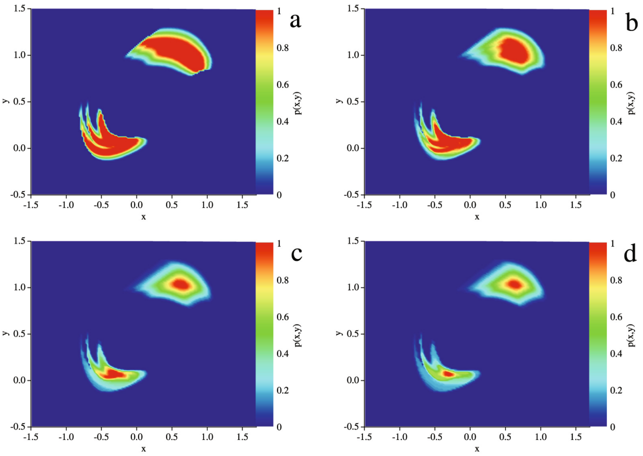


Fig. 3. Transient joint MDFs of displacement x and velocity y for the fuzzy Duffing equation (12) in contour plot at different times. (a) 1T, (b) 2T, (c) 5T, (d) 22T.

ε increases. Fig. 5 shows the evolution of the marginal steady-state MDFs for different intensities of fuzzy noise ε. In Figs. 6 and 7 is shown the evolution of the joint steady-state MDFs for different intensities in the plots of contour and 3D surface respectively. From these figures, it is found that the possibility distribution with two peaks becomes broader and extends toward to B along the unstable manifolds as the intensity of fuzzy noise increases. A merging bifurcation occurs when the possibility distribution merges in through B in the interval $\varepsilon \in (0.07, 0.071)$. The bifurcation results in a sudden change of MDFs from the two peaks of possibility one to one peak band around the unstable manifolds of B. The evolutionary trend of the steady-state MDFs is in accordance with the unstable manifolds expanding toward B as the intensity of fuzzy noise increases.

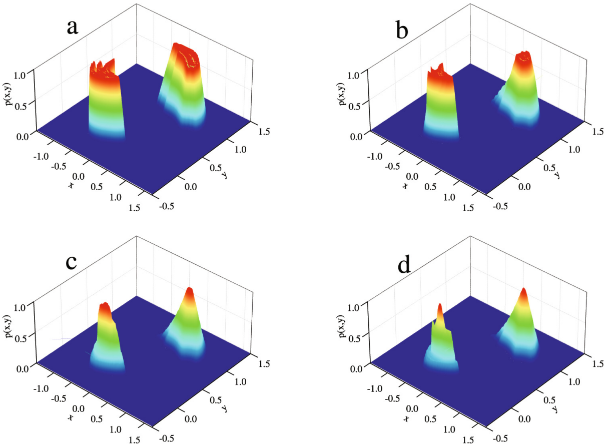


Fig. 4. Transient joint MDFs of displacement x and velocity y for the fuzzy Duffing equation (12) in surface plot at different times. (a) 1T, (b) 2T, (c) 5T, (d) 22T.

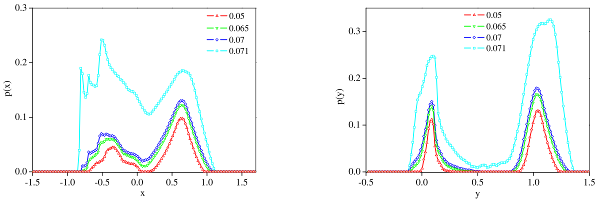


Fig. 5. Steady-state marginal MDFs of displacement x and velocity y for the fuzzy Duffing equation (12) with initial possibility distributions $p_{n_1}(0)$=1 and $p_{n_2}(0)$=1

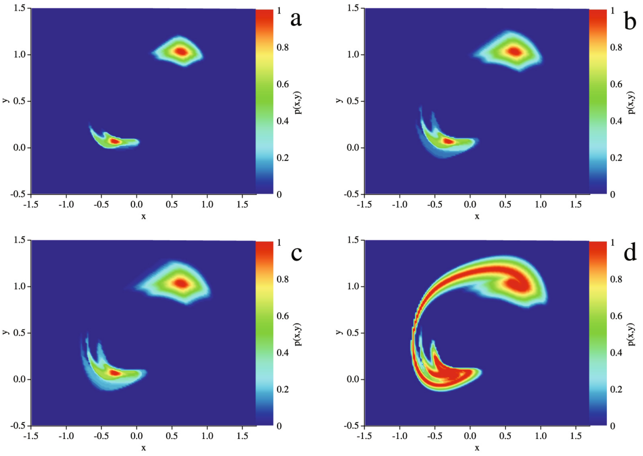


Fig. 6. Joint steady-state MDFs of the fuzzy Duffing system (12) for different intensity of fuzzy noise in contour plot. (a) ε=0.05, (b) ε=0.065, (c) ε=0.07, (d) ε=0.071.

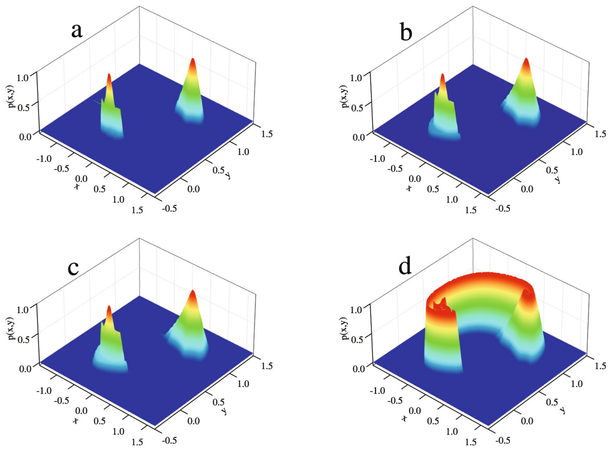


Fig. 7. Joint steady-state MDFs of the fuzzy Duffing system (12) for different intensity of fuzzy noise in 3D surface plot. (a) ε=0.05, (b) ε=0.065, (c) ε=0.07, (d) ε=0.071.

4 Concluding Remarks

The transient and steady-state membership distribution functions (MDFs) of fuzzy response of a forced Duffing oscillator with fuzzy uncertainty are studied by means of the Fuzzy Generalized Cell Mapping (FGCM) method. The FGCM method is first introduced. A rigorous mathematical foundation of the FGCM is established with a discrete representation of the fuzzy master equation for the possibility transition of continuous fuzzy processes. The FGCM offers a very effective approach for solutions to the fuzzy master equation based on the min-max operator of fuzzy logic. Fuzzy response is characterized by its topology in the state space and its possibility measure of MDFs. The response topology is obtained based on the qualitative analysis of the FGCM involving the Boolean operation of 0 and 1. The evolutionary process of transient and steady-state MDFs is determined by the quantitative analysis of the FGCM with the min-max calculations. It is found that the evolutionary orientation of MDFs is in accordance with invariant manifolds leading to invariant sets. In the evolutionary process of a steady-state fuzzy response with an increase of the intensity of fuzzy noise, a merging bifurcation is observed in a sudden change of the MDFs from two sharp peaks of most possibility to one peak band around unstable manifolds. The results obtained herein are of value to real engineering problems. Especially, the evolution of transient and long term steady state MDFs may be difficult to obtain by other conventional methods.

Acknowledgements. This work was supported by the Natural Science Foundation of China through the grants 11332008 and 11172224.

References

[1] Buckley, J.J., Feuring, T.: Fuzzy differential equations. Fuzzy Sets and Systems 110, 43–54 (2000)

[2] Park, J.Y., Han, H.K.: Fuzzy differential equations. Fuzzy Sets and Systems 110, 69–77 (2000)

[3] Yoshida, Y.: A continuous-time dynamic fuzzy system (I) A limit theorem. Fuzzy Sets and Systems 113, 453–460 (2000)

[4] Buckley, J.J., Hayashi, Y.: Applications of fuzzy chaos to fuzzy simulation. Fuzzy Sets and Systems 99, 151–157 (1998)

[5] Friedman, Y., Sandler, U.: Evolution of systems under fuzzy dynamic laws. Fuzzy Sets and Systems 84, 61–74 (1996)

[6] Friedman, Y., Sandler, U.: Fuzzy dynamics as an altemative to statistical mechanics. Fuzzy Sets and Systems 106, 61–74 (1999)

[7] Lin, Y.K., Cai, G.Q.: Probabilistic Structural Dynamics: Advanced Theory and Applications. McGraw-Hill, New York (1995)

[8] Nazaroff, G.J.: Fuzzy topological polysystems. J. Math. Anal. Appl. 41, 478–485 (1973)

[9] Kloeden, P.E.: Fuzzy dynamcal systems. Fuzzy Sets and Systems 7, 275–296 (1982)

[10] Aubin, J.P.: Fuzzy differential inclusions. Problems of Control and Information Theory 19, 55–67 (1990)

[11] Kaleva, O.: The cauchy problem for fuzzy differential equations. Fuzzy Sets and Systems 35, 389–396 (1990)

[12] Ma, M., Friedman, M., Kandel, A.: Numerical solutions of fuzzy differential equations. Fuzzy Sets and Systems 105, 133–138 (1999)

[13] Zhang, Y., Qiao, Z., Wang, G.: Solving processes for a system of first-order fuzzy differential equations. Fuzzy Sets and Systems 95, 333–347 (1998)

[14] Chen, Y.Y., Tsao, T.C.: New approach for the global analysis of fuzzy dynamical systems. In: Proceedings of the 27th IEEE Conference on Decision and Control, Austin, Texas, USA, pp. 1415–1420 (1988)

[15] Chen, Y.Y., Tsao, T.C.: Description of the dynamical behavior of fuzzy systems. IEEE Transactions on Systems, Man and Cybernetics 19(4), 745–755 (1989)

[16] Smith, S.M., Comer, D.J.: Self-tuning of a fuzzy logic controller using a cell state space algorithm. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 6, pp. 445–450 (1990)

[17] Song, F., Smith, S.M., Rizk, C.G.: Optimized fuzzy logic controller design for 4D systems using cell state space technique with reduced mapping error. In: Proceedings of the IEEE International Fuzzy Systems Conference, Seoul, South Korea, vol. 2, pp. 691–696 (1999)

[18] Hsu, C.S.: Cell-to-Cell Mapping: A Method of Global Analysis for Non-linear Systems. Springer, New York (1987)

[19] Edwards, D., Choi, H.T.: Use of fuzzy logic to calculate the statistical properties of strange attractors in chaotic systems. Fuzzy Sets and Systems 88(2), 205–217 (1997)

[20] Sun, J.Q., Hsu, C.S.: Global analysis of nonlinear dynamical systems with fuzzy uncertainties by the cell mapping method. Computer Methods in Applied Mechanics and Engineering 83(2), 109–120 (1990)

[21] Sun, J.Q., Hsu, C.S.: The generalized cell mapping method in nonlinear random vibration based upon short-time Gaussian approximation. Journal of Applied Mechanics 57, 1018–1025 (1990)

[22] 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)

[23] Hong, L., Xu, J.X.: Crises and chaotic transients studied by the generalized cell mapping digraph method. Physics Letters A 262, 361–375 (1999)

[24] Hong, L., Xu, J.X.: Discontinuous bifurcations of chaotic attractors in forced oscillators by generalized cell mapping digraph (GCMD) method. International Journal of Bifurcation and Chaos 11, 723–736 (2001)

[25] Hong, L., Sun, J.Q.: Bifurcations of fuzzy nonlinear dynamical systems. Communications in Nonlinear Science and Numerical Simulation 11(1), 1–12 (2006)

[26] Hong, L., Sun, J.Q.: Codimension two bifurcations of nonlinear systems driven by fuzzy noise. Physica D-Nonlinear Phenomena 213(2), 181–189 (2006)

[27] Hsu, C.S.: Global analysis of dynamical systems using posets and digraphs. International Journal of Bifurcation and Chaos 5(4), 1085–1118 (1995)

[28] Jiang, J., Xu, J.X.: A method of point mapping under cell reference for global analysis of nonlinear dynamical systems. Phys. Lett. A 188, 137–145 (1994)

[29] Jiang, J.: An effective numerical procedure to determine saddle-type unstable invariant limit sets in nonlinear systems. Chin. Phys. Lett. 29(5), 050503 (2012)

Hypervolume Maximization via Set Based Newton's Method

Victor Adrián Sosa Hernández[1], Oliver Schütze[1], and Michael Emmerich[2]

[1] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, C.P. 07360,
Col. San Pedro Zacatenco, Mexico City, México
msosa@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx
[2] Multicriteria Optimization and Decision Analysis Group, LIACS, Leiden University,
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Abstract. The hypervolume indicator is one of the most widely used tool to measure the performance in evolutionary multi-objective optimization. While derivative free methods such as specialized evolutionary algorithms received considerable attention in the past, the investigation of derivative based methods is still scarce. In this work, we aim to make a contribution to fill this gap.

Based on the hypervolume gradient that has recently been proposed for general unconstrained multi-objective optimization problems, we first investigate the behavior of the related hypervolume flow. Under this flow, populations evolve toward a final state (population) whose hypervolume indicator is locally maximal. Some insights obtained on selected test functions explain to a certain extend observations made in previous studies and give some possible insights into the application of mathematical programming techniques to this problem. Further, we apply a population-based version of the Newton Raphson method for the maximization of the hypervolume. Fast set-based convergence can be observed towards optimal populations, however, the results indicate that the success depends crucially on the choice of the initial population.

Keywords: multi-objective optimization, hypervolume indicator, set based optimization, multi-objective gradient, Newton Raphson method.

1 Introduction

The problem of finding a good approximation set to a Pareto front in multiobjective optimization can be recast as a single-objective optimization problem, where the elements of the search space are approximation sets, and the objective function is a unary performance indicator that measures how close the points in a set are converged to and distributed across the Pareto front [1, 2]. Ideally, the performance indicator must not take into account a-priori information of the true Pareto front. The hypervolume indicator [3] is a performance indicator that has these properties and solely requires as a parameter a reference point that bounds the Pareto front from above.

While stochastic, derivative free methods that maximize the hypervolume indicator received considerable attention in the last decade [4–8], research on using deterministic and derivative based methods is still in its infancy.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_2, © Springer International Publishing Switzerland 2014

Set-based hypervolume gradient ascent methods and relay-hybrids with evolutionary methods were proposed in [9] with a focus on the bi-objective case. A full analytical derivation of the hypervolume gradient for more than two objectives and its efficient computation was discussed in [10]. These first results showed that gradient based methods can locally improve accuracy of hypervolume maximal sets, but when started far away from the Pareto front tend to stagnate without reaching the optimum. A full explanation of this behavior is not yet available, nor are there any results on second order methods, such as the Newton Raphson method that promises superlinear or even quadratic convergence rates. A first gradient based memetic strategy to maximize the hypervolume can be found in [11].

The contributions of this paper are two-fold:

1. An investigation of the dynamics and potential problems of gradient based hypervolume maximization is provided by a detailed analysis of its gradient flow. To be more precise, for selected test problems hypervolume maximization will be considered as an initial value problem of the ODE $\dot{P} = \nabla H(P)$, where P denotes the current state of the population and H the hypervolume indicator.
2. A set-based Newton Raphson method for hypervolume maximization is formulated based on the hypervolume gradient and set based Hessian approximation and tested on Pareto fronts with different shapes. This will be the first study on using a second order optimization method for finding Pareto front approximation sets. The findings on this are related to the first study.

The remainder of this paper is organized as follows: In Section 2, we briefly state the background for the understanding of this work. In Section 3, we investigate the hypervolume gradient flow empirically on several test problems. In Section 4, we consider the population based Newton Raphson method for hypervolume maximization. Finally, we draw our conclusions in Section 5 and point out possible paths of future research.

2 Background and Related Work

In the following we consider unconstrained continuous MOPs

$$\min_{x\in\mathbb{R}^n}\{F(x)\}, \tag{MOP}$$

where F is defined as the vector of the objective functions $F : \mathbb{R}^n \to \mathbb{R}^k$, $F(x) = (f_1(x), \ldots, f_k(x))$, and where each objective $f_i : \mathbb{R}^n \to \mathbb{R}$ is (for simplicity) sufficiently smooth. The optimality of a MOP is defined by the concept of *dominance* ([12]): A vector $y \in \mathbb{R}^n$ is *dominated* by a vector $x \in \mathbb{R}^n$ ($x \prec y$) with respect to (MOP) if and only if $f_i(x) \leq f_i(y)$, $i = 1, \ldots, k$, and there exists an index j such that $f_j(x) < f_j(y)$, else y is non-dominated by x. A point $x \in \mathbb{R}^n$ is called *(Pareto) optimal* or a *Pareto point* if there is no $y \in \mathbb{R}^n$ which dominates x. The set of all Pareto optimal solutions is called the *Pareto set*, and is denoted by P. The image $F(P)$ of the Pareto set is called the *Pareto front*. Both sets typically form a ($k-1$)-dimensional object. The hypervolume of a population $H(P)$ is defined as

$$H(P) = \lambda(\cup_{x\in P}[F(x), r]), \tag{1}$$

where λ denotes the Lebesgue measure and r is a user-defined and constant reference point should be dominated by all elements in P [3,5]. By increasing the size of the space that is dominated by a population the hypervolume indicator increases. The hypervolume indicator is a Pareto compliant unary indicator [13]. As a consequence, excepting degenerate cases, maxima of the hypervolume indicator consists of only Pareto optimal elements. Moreover, approximation sets that maximize the hypervolume indicator distribute across the Pareto front with a density of points depending on the deviation of the local slope of the Pareto front from $-45°$ (see [14]). As discussed in [15], a population $\{\mathbf{x}^1, \ldots, \mathbf{x}^\mu\}$ can be expressed as a single vector

$$P := (x_1^{(1)}, \ldots, x_n^{(1)}, \ldots, x_1^{(\mu)}, \ldots, x_n^{(\mu)})^\top \in \mathbb{R}^{\mu \cdot n}. \tag{2}$$

As there are multiple mappings between a set and a population vector, we will consider the one where the μ points are ordered in ascending lexicographical order with respect to the values of F, i. e. a 2-D Pareto front the index of the points will grow from left (1) to right (μ). The hypervolume function on population vectors can now be viewed as a mapping $H : \mathbb{R}^{n\mu} \to \mathbb{R}$. For differentiable population vectors, the gradient field $\nabla H : \mathbb{R}^{n\mu} \to \mathbb{R}^{n\mu}$ and the Hessian $\nabla^2 H : \mathbb{R}^{n\mu} \to \mathbb{R}^{n\mu \times n\mu}$ are defined in the standard way. In [15] it was shown that H is differentiable for population vectors that stem from points sets without duplicate coordinates, with all points in the interior of the reference region $(-\infty, r]$, and with existing Jacobian of F in all points. For sets with duplicate coordinates one sided derivatives are to be taken in some cases. The gradient can be computed by means of the chain rule [15], whereas for the computation of the Hessian matrix so-far only numerical methods using finite differences are available. In the gradient for each point $\mathbf{x}^{(i)}$, $i = 1, \ldots, \mu$ a subgradient of dimension n can be isolated, comprising exactly the partial derivatives for the coordinates of the i-th subvector. It was shown in [15] that this is the gradient of the hypervolume contribution of the i-th point, i. e. $\nabla \mathit{\Delta} H(\mathbf{x}^{(i)})$ with $\mathit{\Delta} H(\mathbf{x}^{(i)}) = H(P) - H(P - \{\mathbf{x}^{(i)}\})$. Subgradients of non-dominated points are positive, unless a stationary point of $\mathit{\Delta} H$ is given, and subgradients of strictly dominated points are zero vectors.

3 Investigating the Hypervolume Flow

First we investigate the flow that is induced by the hypervolume gradient field. This is done in order to see properties to understand the success or failure of gradient based search algorithms applied to this problem. In particular, we will observe a certain 'creepiness' in the movement of some individuals compared to some others (the ones at the boundary and in relative steep/flat regions of the Pareto front). This has the consequence that certain individuals get 'lost' (dominated by other points) under iteration of these set-based algorithms.

Recall from the introduction that after a straightforward re-ordering of the variables we can state the hypervolume function H population-wise (i.e., the variables of H are individuals of a given population). Doing so, we can state the hypervolume flow starting with a given population as the following initial value problem:

$$\begin{aligned} P(0) &= P_0 \subset \mathbb{R}^n \quad \text{with} \quad |P_0| = \mu \\ \dot{P}(t) &= \nabla H(P(t)) \end{aligned} \tag{3}$$

By following this flow, we obtain for every initial population P_0 a final population whose hypervolume is locally optimal (this is due to the fact that the hypervolume gradient is zero at every end point of (3)). Thus, the geometry of such solution curves are of particular interest for the understanding of the success and failure of the related gradient-based numerical optimization techniques.

Within this study, we make our experiments on the following four bi-parameter bi-objective models: The generalized Schaffler problems (GSP) by Emmerich and Deutz ([16])

$$\begin{aligned} f_1 &= \frac{(\sum_{i=1}^n x_i^2)^\alpha}{(n^\alpha)} \\ f_2 &= \frac{(\sum_{i=1}^n (1-x_i)^2)^\alpha}{(n^\alpha)} \end{aligned} \tag{4}$$

The Pareto set of this problem family is given by the line segment connecting the points $(0,0)^T$ and $(1,1)^T$ (including the end points). The Pareto front is concave for $\alpha < 0.5$, linear for $\alpha = 0.5$, and convex for $\alpha > 0.5$. For $\alpha = 0.5$ the hypervolume indicator maximum is known to be an evenly spaced point set on the efficient set and Pareto front [16].

The next model is MOP CONV1 ([17])

$$\begin{aligned} f_1 &= (x_1-1)^4 + (x_2-1)^2 \\ f_2 &= (x_1+1)^2 + (x_2+1)^2 \end{aligned} \tag{5}$$

The Pareto set is a curve connecting the points $m_1 = (1,1)^T$ and $m_2 = (-1,-1)^T$ and its Pareto front is convex.

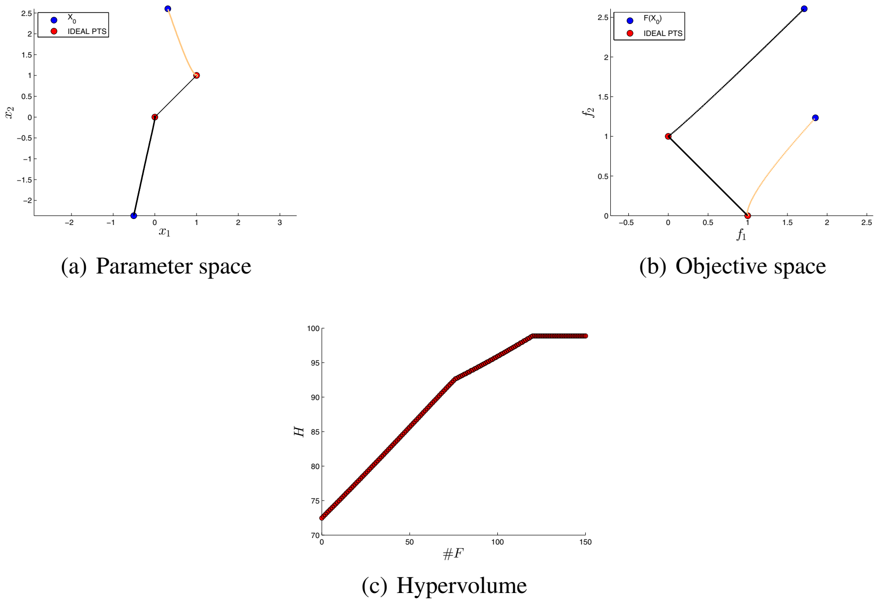
A variant of the above model is CONV2 ([18])

$$\begin{aligned} f_1 &= (x_1-1)^2 + (x_2-1)^2 \\ f_2 &= (x_1+1)^2 + (x_2+1)^2 \end{aligned} \tag{6}$$

The Pareto set is a line segment connecting m_1 and m_2, and the Pareto front is convex. Finally, we will consider the MOP DENT ([19])

$$\begin{aligned} f_1 &= \frac{1}{2} \cdot (\sqrt{(1+(x_1+x_2)^2)} + \sqrt{(1+(x_1-x_2)^2)} + x_1 - x_2) + \lambda \cdot e^{(-1\cdot(x_1-x_2)^2)} \\ f_2 &= \frac{1}{2} \cdot (\sqrt{(1+(x_1+x_2)^2)} + \sqrt{(1+(x_1-x_2)^2)} - x_1 + x_2) + \lambda \cdot e^{(-1\cdot(x_1-x_2)^2)} \\ \lambda &= 0.85 \end{aligned} \tag{7}$$

For the domain $Q = [-4,4]^2$ the Pareto front is the line segment connecting the points $(4,-4)^T$ and $(-4,4)^T$ and its Pareto front is convex-concave.



(a) Parameter space

(b) Objective space

(c) Hypervolume

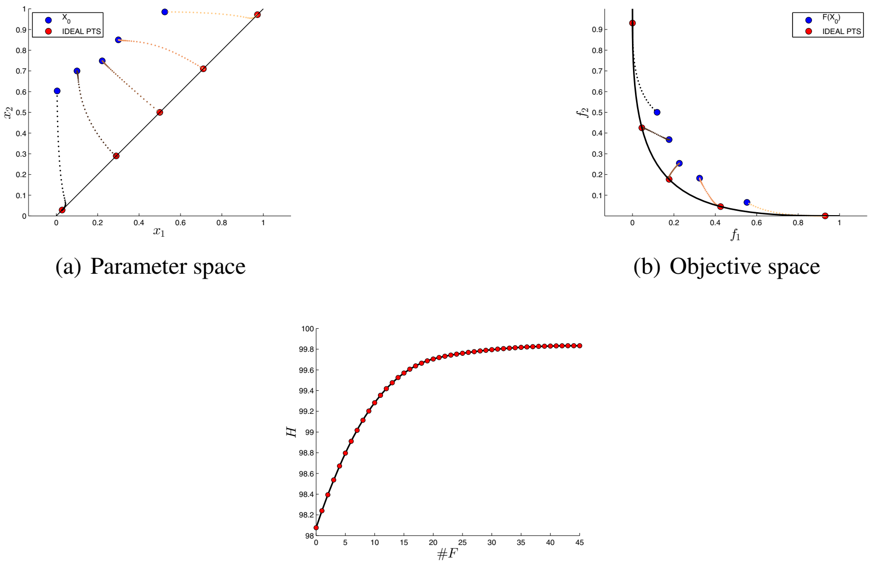
Fig. 1. Hypervolume flow on a 2-element population on MOP GSP for $\alpha = 0.5$

To examine the hypervolume flow we first consider MOP GSP for $\alpha = 0.5$ (see Figure 1). The optimal archive in this case are two Pareto optimal solutions near to the end points of the solution set. Starting with the 2-element archive $P_0 = \{x_1 = (-0.25, -2.5)^T, x_2 = (0, 2.5)^T\}$, we can observe that both individuals x_1 and x_2 directly move toward the nearest optimal solution. The flow hence yields the desired behavior.

As a second positive example we consider a 5-element population on MOP GSP for $\alpha = 1.25$, see Figure 2. Similarly to the above example, all individuals of the initial population perform a more or less direct movement (both in decision and objective space) toward the optimal 5-element HV archive for the reference point $r = (10, 10)^T$.

Next, we consider the same initial population as before, but consider the MOP GSP with $\alpha = 0.5$ (i.e., only the value of α in the model has been changed, the Pareto front is now linear). Instead of a convergent behavior we see in Figure 3 that the extreme solutions x_1 and x_5 (to be more precise, the solutions such that the images $F(x_i)$ are minimal according to f_1 and f_2, respectively) perform a movement toward the extreme points of the Pareto set/front. The other solutions also perform a movement toward the Pareto set, however, it is apparent that this is done with a much slower 'speed'. The reason for this 'creepiness' is certainly the huge difference in the norms of the subgradients: A discretization of (3) via the Euler method leads from a given population P_i to the new one

$$P_{i+1} = P_i + t_i \nabla H(P_i), \tag{8}$$



(a) Parameter space

(b) Objective space

(c) Hypervolume

Fig. 2. Hypervolume flow on a 5-element population on MOP GSP for $\alpha = 1.25$

where $t_i > 0$ is the chosen step size. For every individual $p_j \in P_i$ it thus holds

$$p_j^{new} = p_j + t_i g_j, \tag{9}$$

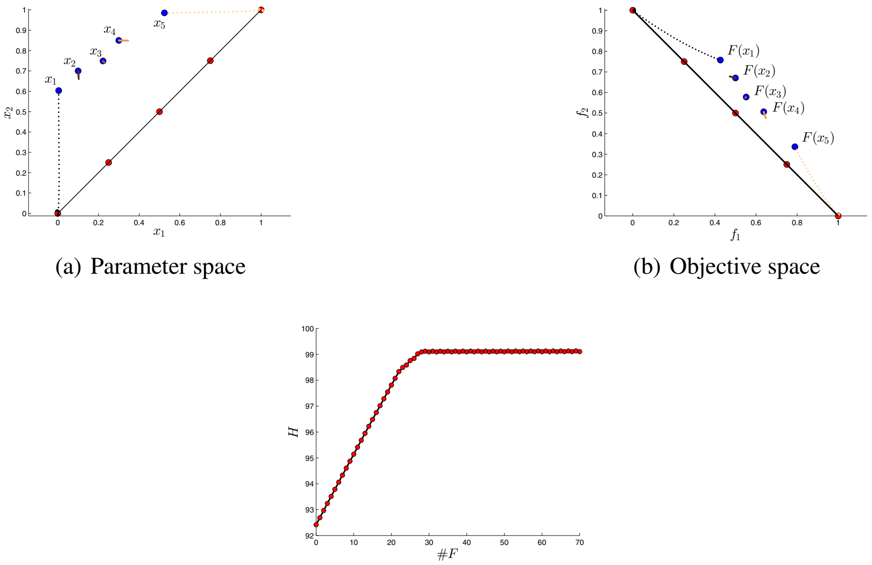
where p_j^{new} denotes the new individual and $g_j \in \mathbb{R}^n$ the subgradient of ∇H and p_i. For the difference of the two individuals it holds hence

$$\|p_j^{new} - p_j\| = t_i \|g_j\|. \tag{10}$$

Since t_i is equal for all subgradients, the speed of the movement is hence entirely determined by the norm of the subgradients. Figure 4 shows the evolution of the norms of the subgradients for the populations considered in Figure 2 and 3. In both cases, there is a significant difference in the norms, namely the norms for the extreme solutions are much higher than the norms of the other individuals. While this difference reduces for the first problem until all norms are in the same range, this does not hold for the second problem. Instead, this difference is nearly stable for the considered sequence.

Figure 5 shows a similar behavior. Here, we have taken the same initial population and the same MOP, but have chosen $\alpha = 0.25$ so that the Pareto front is concave.

Figures 6 and 7 show two examples of 5-element populations where domination occurs. To be more precise, it happens along the gradient flow that solutions in the population start to dominate others. Strictly dominated solutions have a hypervolume contribution that is constantly zero in some epsilon environment and their subgradients

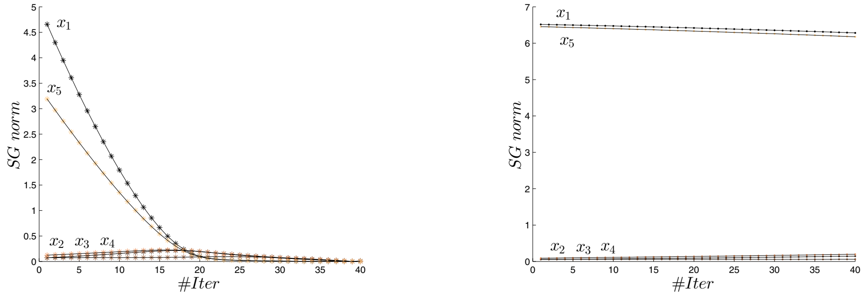


(a) Parameter space

(b) Objective space

(c) Hypervolume

Fig. 3. Hypervolume flow on a 5-element population on MOP GSP for $\alpha = 0.5$ showing a 'creepy' behavior of some individuals



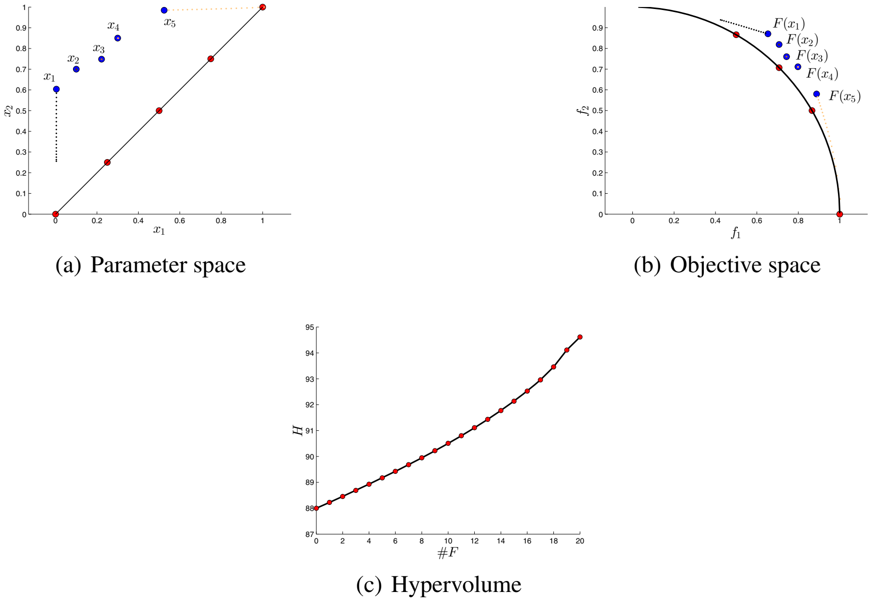
(a) GSP, $\alpha = 1.25$

(b) GSP, $\alpha = 0.5$

Fig. 4. Behavior of the norms of the subgradients of the populations considered in Figures 2 and 3

are zero vectors which causes the points that correspond to these subgradients to become stationary. The search continues with the remaining non-dominated solutions. In both cases, the iterations converge toward the optimal HV archive, albeit for 3 elements.

A study on the use of the steepest ascent method for the numerical realization of (3) can be found in [9]. Linear set-wise convergence has been achieved in case of linear Pareto fronts, however, also the loss of certain elements during the search as a result of domination has been reported. The above observed creepiness may be an explanation for this phenomenon.

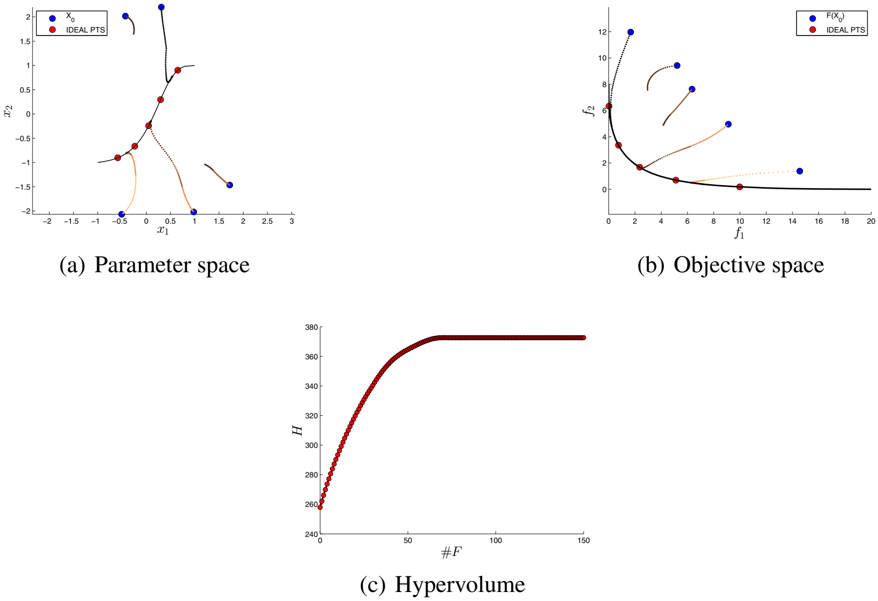


(a) Parameter space

(b) Objective space

(c) Hypervolume

Fig. 5. Hypervolume flow on a 5-element population on MOP GSP for $\alpha = 0.25$ showing a 'creepy' behavior of some individuals

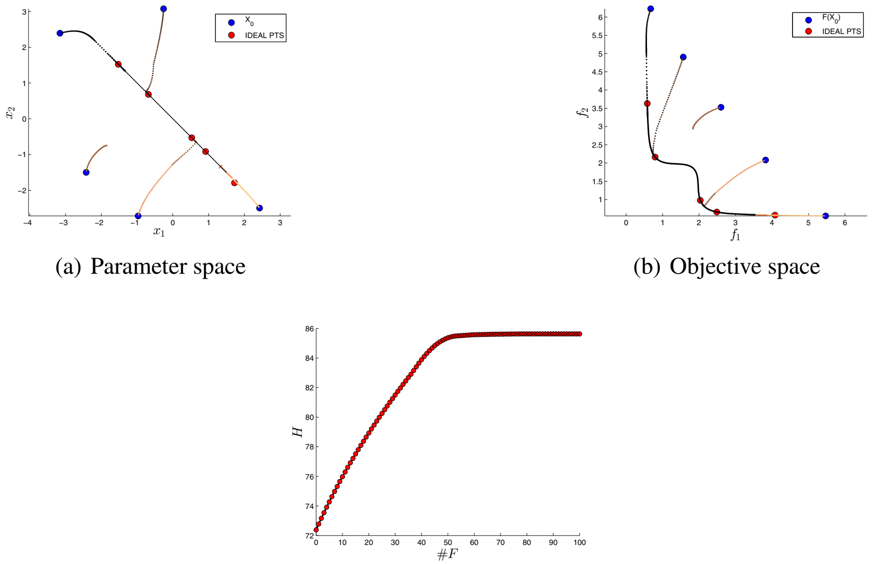


(a) Parameter space

(b) Objective space

(c) Hypervolume

Fig. 6. Hypervolume flow on a 5-element population on MOP CONV1 where domination occurs



(a) Parameter space

(b) Objective space

(c) Hypervolume

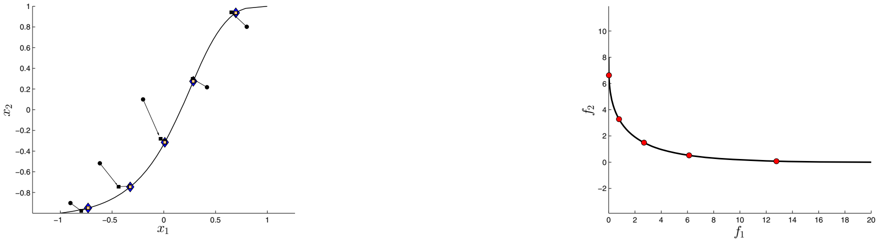
Fig. 7. Hypervolume flow on a 5-element population on MOP DENT where domination occurs

4 Hypervolume Approximations Using the Newton Raphson Method

As a next step it would be interesting if next to gradient based methods also second order methods can be applied successfully on the population based hypervolume indicator function. Here, we make a first attempt and test the Newton Raphson method to obtain populations where the hypervolume gradient vanishes. We stress that the application of the Newton Raphson method is not novel in multi-objective optimization (e.g, the variant proposed in [20]), however, all these methods are point-wise iterative and lead thus from an initial point $x_0 \in \mathbb{R}^n$ to *one* Pareto optimal solution. The consideration of the *entire population* is novel to the authors' best knowledge.

In analogy with (3), we can state the population-based version of the Newton Raphson method for hypervolume maximization as

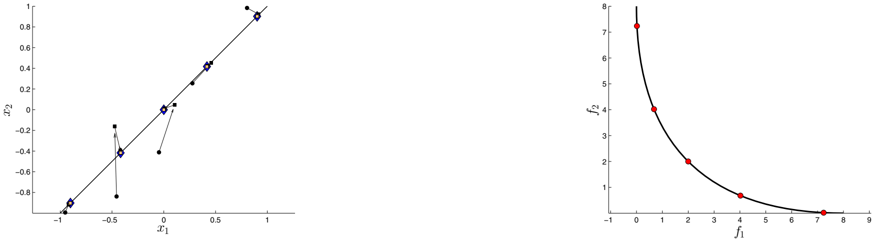
$$\begin{aligned} P_0 \subset \mathbb{R}^n \quad \text{with} \quad |P_0| = \mu \\ P_{i+1} = P_i - \nabla^2 H(P_i)^{-1} \nabla H(P_i), \quad i = 0, 1, 2, \ldots \end{aligned} \tag{11}$$



(a) Parameter space

(b) Objective space

Fig. 8. Application of the Newton Raphson method on a 5-element population on MOP CONV1 yielding quadratic convergence



(a) Parameter space

(b) Objective space

Fig. 9. Application of the Newton Raphson method on a 5-element population on MOP GSP yielding quadratic convergence

Figures 8 and 9 show two examples of initial populations with 5 elements that converge quadratically to the optimal solution. The convergence properties are demonstrated in Tables 1 and 2, respectively. In both tables, a comparison is made to the population-based steepest ascent method yielding a different convergence rate.

Figure 10 shows another example of an initial 5-element archive. Also in this case we a observe quadratic convergence rate, however, in this case to the optimal archive with 3 elements. The reason is that during the iteration two individuals are generated that are dominated by others in the population and thus fixed in their position in subsequent steps.

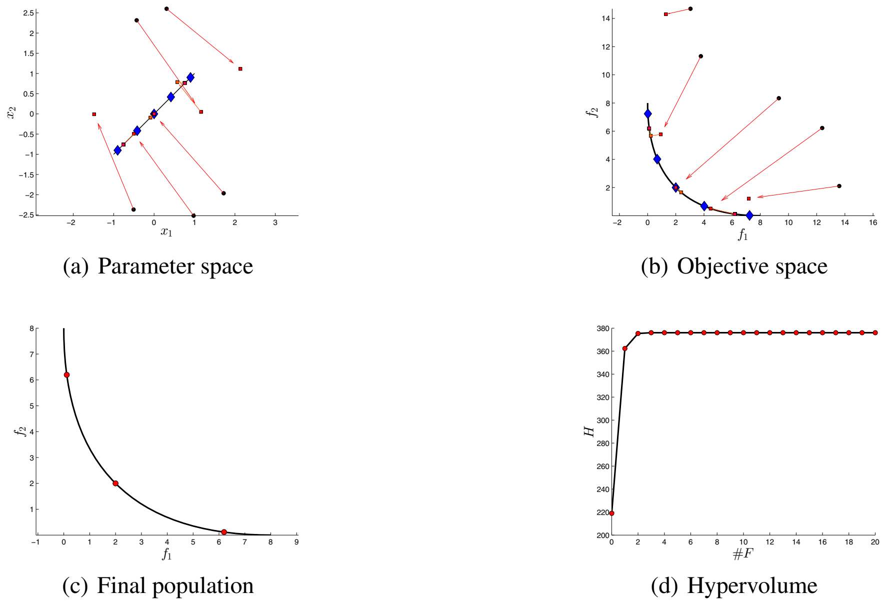
Further empirical studies have confirmed a similar behavior: Quadratic convergence could be observed in all cases, but the loss of individuals due to dominance was not uncommon. So far, we do not have a detailed explanation or indicator for this phenomenon, and have to leave this for future work. If the initial population was close to the optimal one, however, convergence is sure which is a known fact from the Newton Raphson method. We think that step size control, maybe even considered separately for each individual, might relax the problem, but have to leave this as well for future research.

Table 1. Errors of the iterations obtained by the Newton Raphson (RN) and the steepest ascent method (StA) on MOP CONV1 (refer to Figure 8)

Iteration	Error RN	Error StA
0	6.3019173538212	6.3019173538212
1	0.3822821410060	1.1256506467367
2	0.0036900401779	0.6345590706760
3	0.0000002871523	0.4247608492216
4	0.0000000000001	0.3043120967485
5	0.0	0.2237221272231
⋮	⋮	⋮
45	0.0	0.0000099377847
46	0.0	0.0000077289906
47	0.0	0.0000060108811
48	0.0	0.0000046743939
49	0.0	0.0000036350798
50	0.0	0.0000028266895

Table 2. Errors of the iterations obtained by the Newton Raphson (RN) and the steepest ascent method (StA) on MOP GSP (refer to Figure 9)

Iteration	Error RN	Error StA
0	1.548406126519012	1.5484061265190
1	0.462396554222039	0.5448676870680
2	0.000049417729375	0.2211324621996
3	0.000000013384221	0.0937296798889
4	0.000000000000001	0.0400550587705
5	0.0	0.0177895807373
⋮	⋮	⋮
45	0.0	0.0000000000062
46	0.0	0.0000000000037
47	0.0	0.0000000000022
48	0.0	0.0000000000013
49	0.0	0.0000000000007
50	0.0	0.0000000000002



(a) Parameter space

(b) Objective space

(c) Final population

(d) Hypervolume

Fig. 10. Application of the Newton Raphson method on a 5-element population on MOP CONV2 yielding quadratic convergence. We obtain quadratic convergence toward the optimal archive with 3 elements.

5 Conclusions and Future Work

In this work, we have made a first attempt to understand the dynamical behavior of population-based and gradient-based hypervolume maximization. For this, we have first investigated the hypervolume flow as an initial value problem of the ODE $\dot{P} = \nabla H(P)$, where $\nabla H(P)$ denotes the hypervolume gradient. This was done since solutions of the related initial value problems lead to populations whose hypervolumes are locally optimal. One particular problem of the hypervolume gradient is that individuals have a zero subgradient if they are strictly dominated by other elements in the population. Once domination occurs this dominated element will not be considered in subsequent iterations. If so, the population will still evolve toward a final population that is locally optimal, however, that contains fewer elements in the population which is not desired. In that case, apparently the globally optimal population cannot be reached since the full potential of the population is not tapped.

By considering several test examples we have observed a certain ‘creepiness’ in the flow coming from the fact that the norms of the subgradients (which are related to the hypervolume contributions of the individual points and the local slope of the Pareto front) are in different ranges. Such a creeping behavior led in some cases to domination of points in the population. Thus, it is not sufficient for an initial population that its individuals are mutually non-dominated. Further, it seems to be beneficial for the search dynamics, if the norms of the subgradients are in the same range.

Second, we have applied the Newton Raphson method on the population-based hypervolume function. We have observed a convergent behavior in all cases indicating quadratic convergence rates. However, we have as well observed convergence toward an optimal population with smaller numbers of individuals. This is also related to the creepiness causing points getting dominated as observed for the hypervolume flow.

Though the first results, in particular the observation of a quadratic convergence rate, are very promising, there are many things to be done to extend this initial study. There is, for instance, the further investigation of the Newton Raphson method including the analytical derivation of the Hessian, the related convergence analysis as well as the consideration of higher dimensions (both n and k). Further, in consideration of the creepiness it might be worth discussing optimization techniques that utilize different step sizes for each individual depending on the norms of the subgradients as e.g. done in [11] in a similar context, or introducing non-zero gradients for dominated space by means of penalty terms as suggested in [9]. Finally, since the proper choice of the initial population is a delicate task, this might be done by another algorithm. The development of a specialized memetic algorithm (i.e., evolutionary algorithm coupled with the Newton Raphson method as local search) seems to be most promising and will be addressed by the authors in the next step.

Acknowledgement. The first author acknowledges support from CONACyT through a scholarship to pursue undergraduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 128554. The third author acknowledges support from the DELIVER project, EUREKA/Agentschap NL.

References

[1] Fleischer, M.: The measure of pareto optima applications to multi-objective metaheuristics. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 519–533. Springer, Heidelberg (2003)

[2] Zitzler, E., Thiele, L., Bader, J.: SPAM: Set preference algorithm for multiobjective optimization. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 847–858. Springer, Heidelberg (2008)

[3] Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, ETH Zurich, Switzerland (1999)

[4] Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)

[5] Emmerich, M., Beume, N., Naujoks, B.: An emo algorithm using the hypervolume measure as selection criterion. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 62–76. Springer, Heidelberg (2005)

[6] Igel, C., Hansen, N., Roth, S.: Covariance matrix adaptation for multi-objective optimization. Evol. Comput. 15(1), 1–28 (2007)

[7] Koch, P., Kramer, O., Rudolph, G., Beume, N.: On the hybridization of sms-emoa and local search for continuous multiobjective optimization. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 603–610. ACM (2009)

[8] Bader, J., Zitzler, E.: Hype: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation 19(1), 45–76 (2011)

[9] Emmerich, M., Deutz, A., Beume, N.: Gradient-based/evolutionary relay hybrid for computing pareto front approximations maximizing the s-metric. In: Bartz-Beielstein, T., Blesa Aguilera, M.J., Blum, C., Naujoks, B., Roli, A., Rudolph, G., Sampels, M. (eds.) HCI/ICCV 2007. LNCS, vol. 4771, pp. 140–156. Springer, Heidelberg (2007)

[10] Emmerich, M., Deutz, A.: Time complexity and zeros of the hypervolume indicator gradient field. In: Schuetze, O., Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Moral, P.D., Legrand, P. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III. SCI, vol. 500, pp. 169–193. Springer, Heidelberg (2014)

[11] Hernández, V.A.S., Schütze, O., Rudolph, G., Trautmann, H.: The directed search method for pareto front approximations with maximum dominated hypervolume. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 189–205. Springer, Heidelberg (2013)

[12] Pareto, V.: Manual of Political Economy. The MacMillan Press (1971 (original edition in French in 1927))

[13] Zitzler, E., Thiele, L., Laumanns, M., Foneseca, 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)

[14] Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Theory of the hypervolume indicator: optimal mu-distributions and the choice of the reference point. In: FOGA 2009: Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms, pp. 87–102. ACM, New York (2009)

[15] Emmerich, M., Deutz, A. (eds.): EVOLVE 2013 - A Bridge Between Probability. Set Oriented Numerics, and Evolutionary Computation - Short Paper and Extended Abstract Proceedings, LIACS, Leiden University (July 2013)

[16] Emmerich, M., Deutz, A.: A family of test problems with pareto-fronts of variable curvature based on super-spheres. In: MCDM 2006, Chania, Greece, June 19-23 (2006)

[17] Schütze, O.: Set Oriented Methods for Global Optimization. PhD thesis, University of Paderborn (2004), http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/

[18] Köppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)

[19] Witting, K.: Numerical Algorithms for the Treatment of Parametric Optimization Problems and Applications. PhD thesis, University of Paderborn (2012)

[20] Fliege, J., Drummond, L.M.G., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM J. on Optimization 20, 602–626 (2009)

Part II

Computational Game Theory

On Values of Games

Ahmad Termimi Ab Ghani[1] and Kojiro Higuchi[2]

[1] School of Informatics and Applied Mathematics
Universiti Malaysia Terengganu 21030, Malaysia
termimi@umt.edu.my
[2] Department of Mathematics and Informatics
Faculty of Science, Chiba University, Japan
khiguchi@g.math.s.chiba-u.ac.jp

Abstract. In this paper, we present some recent results of infinite games played on a finite graph. We mainly work with generalized reachability games and Büchi games. These games are two-player concurrent games in which each player chooses simultaneously their moves at each step. We concern here with a description of winning strategies and payoff functions over infinite plays. Each play and the outcome of a game are completely determined by strategies of the players. We classify strategies regarding their use of history. Our goal is to give simple expressions of values for each game. Moreover, we are interested in the question of what type of optimal (ϵ-optimal) strategy exists for both players depending on the type of games.

Keywords: reachability games, Büchi games, optimal strategy.

1 Introduction

We consider two-player simultaneous games played on finite graphs. For each round of the game, Player I and Player II choose their actions simultaneously and then the next state is determined. A finite or infinite sequence of state obtained are the result of the play. We investigate on generalized reachability games whose payoff functions can be described as a label function on the set of states over the non-negative real numbers. We mainly focus on Büchi games where the Player I want to visit target states infinitely many times and the Player II want to prevent from reaching the target states infinitely often. These are zero-sum games, and the reachability objective is one of the most basic objectives among the Borel hierarchy. Since there are two players on whose decisions the probability depends, we talk about the highest probability that the Player I can achieve against any opponent's strategy. Similarly, we also discuss the lowest probability that the Player II can achieve against any strategy of Player I. If these two quantities are equal, we call them the value of the game and say the game is determined. An optimal strategy for Player I is a strategy that guarantees the value of the game from each position. The way to determine the winner is called a winning objective. It is a set of infinite plays that we define as a winning for the Player I. In this study, we give answers to the following questions. Are the games determined or can we derive the value of game? Is it possible to show optimal and ϵ-optimal strategies in some way?

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_3, © Springer International Publishing Switzerland 2014

Although Martin's theorem showed that every Borel game is determined, our results provide a specific proof for these types of games and may give more insight into this area, especially games on graphs.

1.1 Related Works and Motivation

In 1953, Gale and Stewart [10] introduced the general theory of infinite games, called Gale-Stewart games, which are two-player infinite games with perfect information. The theory of Gale-Stewart games has been investigated by many mathematicians and logicians, and until now it is one of the interesting topics in game theory and mathematical logic. This game is an infinite zero-sum game with perfect information because one of the players always wins and the other losses and the game is played in turn. The determinacy results for turn-based games with Borel objective was established by a deep result of Martin [14]. He proved that under some fairly general assumptions, one player has a winning strategy. On the other hand, the determinacy for one-round simultaneous games was proven by von Neumann [15] using his famous minmax theorem. Infinite versions of von Neumann's games were introduced by David Blackwell [1]. The determinacy for such games with Borel objective was established by Martin [13]. The proof of Martin's theorem is the culmination of a long series of results proving the determinacy of games of increasing Borel hierarchy.

Recently, Jan Krcăl in [11] studied a determinacy of stochastic turn-based games focused on some winning objectives. Turn-based stochastic games are infinitely long sequential games with perfect information played by two players and a random player. He mainly discussed the reachability games and showed that the games are determined whether the games are finite or infinite. He also proved the existence of an optimal memoryless and deterministic strategy in the finite Büchi games. There are still many challenging open problems in the area of turn-based stochastic games. The existing results about infinite-state games usually concern on Markov Decision Process (MDPs) [12]. Moreover, many of the fundamental results are still waiting to be discovered in the infinite games with imperfect information, especially their use of payoff functions. Over the games on graph, the typical and most studied payoff functions are the limit-average (also called mean-payoff) and the discounted sum of the rewards along the path. To know the definition of mean payoff functions for example see [9], [8], and [16]; discounted payoff for details see also [9] and [6]. Besides their simple definitions, these two payoff functions enjoy the property that memoryless optimal strategies always exist, especially in turn-based stochastic games. In [2], they introduced a multi-mean payoff on turn-based stochastic parity games. This work can be seen as an extension of [3] where mean-payoff parity games have been studied. While Chatterjee et al. [4] defined another simple payoff functions which contain both the limit-average and the discounted sum functions in two-player turn-based games on a graph. In our study, a labelling function defined in generalized reachability games can be seen as a weighted reachability payoff function assigns to every infinite play either 0 if the game does not visit a target state, or the reward (positive real number) of the first target state visited by the player.

This paper is organized as follows. We first introduce the terminology of games, strategies and values. We then study a generalized reachability game and described its values as limits of finite-step games. Our main contribution is showing the existence of memoryless and randomized ε-optimal strategy for Player I, that is, the strategy in which depend only on the current state, satisfies the objective with probability within an ε difference of the value of the game. We then turn to games with Büchi objectives and use the results of generalized rachability games to show the value of Büchi games can be approximated in some way.

2 Games

This section gives preliminaries that is necessary for understanding the argument presented in subsequent sections.

Definition 1. *A (two-player simultaneous infinite) game is a quadruple $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$, where S, A_{I} and A_{II} are nonempty finite sets and δ is a function from $S \times A_{\mathrm{I}} \times A_{\mathrm{II}}$ into S. Elements of S are called states. Elements of A_{I} are called actions or moves of Player I. Similarly, elements of A_{II} are called actions or moves of Player II. δ is called a transition function.*

Definition 2. *A path or a play of a game $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ is a finite or infinite sequence $s_0 s_1 s_2 ...$ of states in S such that for all $n \in \mathbb{N}$, there exist $a_n \in A_{\mathrm{I}}$ and $b_n \in A_{\mathrm{II}}$ where $\delta(s_n, a_n, b_n) = s_{n+1}$. Infinite paths of $\mathbb{G}$ are sometimes called runs. We write $\Omega(\mathbb{G})$ for the set of all infinite plays; and $\Omega^{\mathrm{fin}}(\mathbb{G})$ for the set of all finite plays of non-zero length. Sometimes we write Ω or Ω^{fin} instead of $\Omega(\mathbb{G})$ or $\Omega^{\mathrm{fin}}(\mathbb{G})$ when $\mathbb{G}$ is clear from the context.*

Intuitively, given a game $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$, a function $F : \Omega(\mathbb{G}) \to [0, 1]$ and a state $s \in S$, we imagine the following infinite game $\mathbb{G}_s(F)$: at stage $n \in \mathbb{N} \setminus \{0\}$, we have the finite play $w \restriction n$ with $w(0) = s$, and each player selects their actions $a_{\mathrm{I}} \in A_{\mathrm{I}}$ and $a_{\mathrm{II}} \in A_{\mathrm{II}}$, simultaneously, and, then, the next state $w(n) = \delta(w(n-1), a_{\mathrm{I}}, a_{\mathrm{II}})$ is determined. In this case the value of the play w is $F(w)$. We assume that Player I wants to maximize the value, whereas Player II wants to minimize. For a subset X of $\Omega(\mathbb{G})$, the infinite game $\mathbb{G}_s(X)$ is defined in the same way considering X as its characteristic function. Thus, in the case of a set X instead of a function F, Player I wants to put w into X, whereas Player II wants to avoid it.

The notion of strategies for infinite games plays an important role. Informally, a strategy for a player in the game is a rule that specifies the next move of the player for a given finite play.

For a set A, a *probability distribution* on A is a function $\mu : A \to [0, 1]$ with $\sum_{a \in A} \mu(a) = 1$. We use $\mathcal{D}(A)$ for the set of all probability distributions on A.

Definition 3. *Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game. A (randomized) strategy of Player I in $\mathbb{G}$ is any function $\sigma : \Omega^{\mathrm{fin}}(\mathbb{G}) \to \mathcal{D}(A_{\mathrm{I}})$. We write $\Sigma_{\mathrm{I}}^{\mathbb{G}}$ or Σ_{I} for the set of all strategies of Player I. Similary, a (randomized) strategy of Player II in $\mathbb{G}$ is any function $\tau : \Omega^{\mathrm{fin}}(\mathbb{G}) \to \mathcal{D}(A_{\mathrm{II}})$, and we write $\Sigma_{\mathrm{II}}^{\mathbb{G}}$ or Σ_{II} for the set of all strategies of Player II.*

Definition 4. *Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game. A strategy σ of Player I is called memoryless if $\sigma(p) = \sigma(q)$ holds whenever $p, q \in \Omega^{\mathrm{fin}}(\mathbb{G})$ satisfy $p(|p|-1) = q(|q|-1)$. A memoryless strategy of Player II is defined similarly. We write $\Sigma_{\mathrm{I}}^{\mathrm{M}}$ and $\Sigma_{\mathrm{II}}^{\mathrm{M}}$ for the set of all memoryless strategies of Player I and Player II, respectively.*

Intuitively, for a given finite play, memoryless strategies give the next action depending on the current state rather than depending on the finite play.

Clearly, given a memoryless strategy $\sigma \in \Sigma_{\mathrm{I}}^{\mathrm{M}}$, there exists *the* function $\sigma' : S \to \mathcal{D}(A_{\mathrm{I}})$ such that $\sigma(ps) = \sigma'(s)$ holds for any $ps \in \Omega^{\mathrm{fin}}(\mathbb{G})$ with $s \in S$. We sometimes identify σ with σ'. Similar identification will be used for Player II.

A pair $(\sigma, \tau) \in \Sigma_{\mathrm{I}} \times \Sigma_{\mathrm{II}}$ and a state $s \in S$ determine a probability measure $P_s^{\sigma,\tau}$ on $\Omega_s = \{w \in \Omega : w(0) = s\}$ as follows.

Definition 5. *Let $G = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game. For a pair $(\sigma, \tau) \in \Sigma_{\mathrm{I}}^{\mathbb{G}} \times \Sigma_{\mathrm{II}}^{\mathbb{G}}$ of strategies and a state $s \in S$, $P_s^{\sigma,\tau}$ denotes the probability measure on Ω_s determined by*

$$P_s^{\sigma,\tau}([p]) = \prod_{n \in \{1,\cdots,|\rho|-1\}} \sum \{\sigma(p \upharpoonright n)(a)\tau(p \upharpoonright n)(b) : (p(n-1), a, b) \in \delta^{-1}(p(n))\}$$

for any $p \in \Omega_s^{\mathrm{fin}} = \{q \in \Omega^{\mathrm{fin}} : q(0) = s\}$, where $[p] = \{w \in \Omega : p \subset w\}$.

Intuitively, for a function $F : \Omega \to [0, 1]$ with $P_s^{\sigma,\tau}(F) = \int_{\Omega_s} F dP_s^{\sigma,\tau}$ exists, $P_s^{\sigma,\tau}(F)$ means the expectated value of an infinite game $\mathbb{G}_s(F)$ when Player I and Player II use the strategy σ and τ, respectively. In the case of a subset X of Ω instead of F, $P_s^{\sigma,\tau}(X)$ means the probability that the infinite play in Ω_s belongs to X when Player I and Player II use the corresponding strategies.

Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game, and let $F : \Omega(\mathbb{G}) \to [0, 1]$ satisfy that $P_s^{\sigma,\tau}(F)$ exists for any $\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}$, $\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}$ and $s \in S$. We call such a function F a *payoff function* of $\mathbb{G}$. (In the game $\mathbb{G}(X)$, the set X with such a property is called a *winning set* of $\mathbb{G}$.) The value of Player I in a game $\mathbb{G}_s(F)$ for a state s is the supremum of expected value which Player I can ensure. Formally, it is $\sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F)$. Let $\neg F$ be a function defined by $\neg F(w) = 1 - F(w)$. The value of Player II is defined as $\sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} P_s^{\sigma,\tau}(\neg F)$. This value is equal to $1 - \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} \sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F)$. We say that the game $\mathbb{G}(F)$ is *determinate* if

$$\sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F) + \sup_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} \inf_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} P_s^{\tau,\sigma}(\neg F) = 1$$

holds for any $s \in S$. Or equivalently, the game $\mathbb{G}(F)$ is determinate if and only if

$$\sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F) = \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} \sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F)$$

holds for any $s \in S$. In this case, we write $\mathrm{val}_s^{\mathbb{G}}(F)$ or $\mathrm{val}_s(F)$ instead of $\sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F)$, and call it the *value* at s in the game $\mathbb{G}(F)$.

The following is well-known theorem obtained by Martin.

Theorem 1 (Martin [13]). *Let* $\mathbb{G}$ *be a game and let* $F : \Omega(\mathbb{G}) \to [0,1]$ *a Borel measurable function. Then the game* $\mathbb{G}(F)$ *is determinate.* □

Actually, the function F is Borel measurable function for any game $\mathbb{G}(F)$ studied in this paper later. Thus any game studied in this paper is determinate.

Definition 6. *Let* $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$, $F : \Omega \to [0,1]$ *and* $\epsilon \in [0,1]$. *Suppose that* $\mathbb{G}(F)$ *is determinate. A strategy* $\sigma \in \Sigma_{\mathrm{I}}$ *of Player I is* ϵ*-optimal if* $\inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F) \geq \mathrm{val}_s(F) - \epsilon$ *holds for any* $s \in S$. *Similarly, a strategy* $\tau \in \Sigma_{\mathrm{II}}$ *of Player II is* ϵ*-optimal if* $\sup_{\sigma \in \Sigma_{\mathrm{I}}^{\mathbb{G}}} P_s^{\sigma,\tau}(F) \leq \mathrm{val}_s(F) + \epsilon$ *holds for any* $s \in S$. *Optimal strategies are* 0*-optimal strategies.*

By the definitions a strategy $\sigma \in \Sigma_{\mathrm{I}}$ of Player I is optimal if and only if $\inf_{\tau \in \Sigma_{\mathrm{II}}} P_s^{\sigma,\tau}(F) = \mathrm{val}_s(F)$ holds for all $s \in S$, and $\tau \in \Sigma_{\mathrm{II}}$ is optimal if and only if $\sup_{\sigma \in \Sigma_{\mathrm{I}}} P_s^{\sigma,\tau}(F) = \mathrm{val}_s(F)$ holds for all $s \in S$.

When $\mathbb{G}(F)$ is determinate and ε is a positive real number, then ε-optimal strategies of Player I and Player II always exist by the definition. However, there are some cases that Player I or Player II has no optimal strategy.

Let $\mathbb{G} = (X, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game and let $V : S \to [0,1]$. We define $F_V : \Omega(\mathbb{G}) \to [0,1]$ by $F_V(w) = V(w(1))$. Games of the form $\mathbb{G}(F_V)$ are called *one-step games*. We write $\mathbb{G}(V)$ meaning $\mathbb{G}(F_V)$, and we write $\mathrm{val}_s(V)$ for $s \in S$ instead of $\mathrm{val}_s(F_V)$. In one-step games optimal strategies always exist for each player. This theorem is well-known as von Neumann's minmax theorem.

Theorem 2 (von Neumann [15]). *In any one-step game, both players have their optimal strategies.* □

3 Generalized Reachability Games

Reachability games are in some respect the simplest infinite games. We will prove some basic facts on a generalized version of reachability games, called generalized reachability games. We will describe the value of reachability games as a limit value of finite-step games. We will see that Player II has a memoryless optimal strategy, and Player I has a memoryless ε-optimal strategy in any generalized reachability games for any positive real number ε. Nevertheless, in general it is known that, even in a reachability game, Player I may not have an optimal strategy.

Definition 7. *Let* $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ *be a game. A function* ℓ *is called a label on* S *if* $\mathrm{dom}(\ell) \subset S$ *and* $\ell(s) \in [0,1]$ *for any* $s \in \mathrm{dom}(\ell)$. *We define* $\mathcal{R}^{\mathbb{G},\ell} : \Omega(\mathbb{G}) \to [0,1]$ *by*

$$\mathcal{R}^{\mathbb{G},\ell}(w) = \begin{cases} \ell(w(N_w)) & \textit{if } (\exists N \in \mathbb{N})[w(N) \in \mathrm{dom}(\ell)], \\ 0 & \textit{otherwise}, \end{cases}$$

where N_w *is the least natural number* N *such that* $w(N) \in \mathrm{dom}(\ell)$. *A game of the form* $\mathbb{G}(\mathcal{R}^{\mathbb{G},\ell})$ *is called a generalized reachability game.*

For a subset T of S, let $\mathcal{R}^{\mathbb{G},T} = \mathcal{R}^{\mathbb{G},\ell_T}$, where $\ell_T : T \to \{1\}$. Games of the form $\mathbb{G}(\mathcal{R}^{\mathbb{G},T})$ are called *reachability games*.

Definition 8. *Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game and let ℓ a label on S. For every state $s \in S$ and $n \in \mathbb{N}$, we define $V_n^{\mathbb{G},\ell} : S \to [0,1]$ inductively by*

$$V_0^{\mathbb{G},\ell}(s) = \begin{cases} \ell(s) & \textit{if } s \in \mathrm{dom}(\ell), \\ 0 & \textit{otherwise}, \end{cases} \qquad V_{n+1}^{\mathbb{G},\ell}(s) = \begin{cases} \ell(s) & \textit{if } s \in \mathrm{dom}(\ell), \\ \mathrm{val}_s(V_n^{\mathbb{G},\ell}) & \textit{otherwise}. \end{cases}$$

We let $V^{\mathbb{G},\ell}(s) = \lim_{n\to\infty} V_n^{\mathbb{G},\ell}(s)$ for any state s, and we call it the limit value at s.

For a label ℓ on S and $n \in \mathbb{N}$, we define $\mathcal{R}_n^{\mathbb{G},\ell} : \Omega(\mathbb{G}) \to [0,1]$ by $\mathcal{R}_n^{\mathbb{G},\ell}(w) = s_w$ if there exists $m \leq n$ with $w(m) \in \mathrm{dom}(\ell)$ and $\mathcal{R}_n^{\mathbb{G},\ell}(w) = 0$ otherwise.

Theorem 3. *For any $n \in \mathbb{N}$, both players have their optimal strategy in the game $\mathbb{G}(\mathcal{R}_n^{\mathbb{G},\ell})$, and the equality $V_n^{\mathbb{G},\ell}(s) = \mathrm{val}_s(\mathcal{R}_n^{\mathbb{G},\ell})$ holds for all $s \in S$.*

Proof. We define σ_n^* and τ_n^* inductively. Let σ_0^* and τ_0^* be any strategies. Now suppose that we have constructed σ_n^* and τ_n^*. Choose σ and τ as optimal strategies of Player I and II respectively in the one-step game $\mathbb{G}(V_n^{\mathbb{G},\ell})$. Define σ_{n+1}^* by $\sigma_{n+1}^*(s) = \sigma(s)$ and $\sigma_{n+1}^*(s\rho) = \sigma_n^*(\rho)$ for any $s \in S$ and any $\rho \neq \emptyset$ with $s\rho \in \Omega^{\mathrm{fin}}$. Similarly, define τ_{n+1}^* by $\tau_{n+1}^*(s) = \tau(s)$ and $\tau_{n+1}^*(s\rho) = \tau_n^*(\rho)$ for any $s \in S$ and any $\rho \neq \emptyset$ with $s\rho \in \Omega^{\mathrm{fin}}$. It is easy to see by induction on n that σ_n^* and τ_n^* satisfy the equalities $V_n^{\mathbb{G},\ell}(s) = \inf_{\tau\in\Sigma_{\mathrm{II}}} P_s^{\sigma_n^*,\tau}(\mathcal{R}_n^{\mathbb{G},\ell}) = \sup_{\sigma\in\Sigma_{\mathrm{I}}} P_s^{\sigma,\tau_n^*}(\mathcal{R}_n^{\mathbb{G},\ell})$. This equalities imply that the σ_n^* and τ_n^* are optimal strategies in the game $\mathbb{G}(\mathcal{R}_n^{\mathbb{G},\ell})$ and $V_n^{\mathbb{G},\ell}(s) = \mathrm{val}_s(\mathcal{R}_n^{\mathbb{G},\ell})$ holds. □

Now we verify the value $\mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell})$ is equivalent to the limit value $V^{\mathbb{G},\ell}(s)$.

Theorem 4. *For any state $s \in S$, the equation $V^{\mathbb{G},\ell}(s) = \mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell})$ holds.*

Proof. It is enough to show that the following inequalities:

$$\inf_{\tau\in\Sigma_{\mathrm{II}}} \sup_{\sigma\in\Sigma_{\mathrm{I}}} P_s^{\sigma,\tau}(\mathcal{R}^{\mathbb{G},\ell}) \leq V^{\mathbb{G},\ell}(s) \leq \sup_{\sigma\in\Sigma_{\mathrm{I}}} \inf_{\tau\in\Sigma_{\mathrm{II}}} P_s^{\sigma,\tau}(\mathcal{R}^{\mathbb{G},\ell}).$$

To show the first inequality, choose an optimal strategy τ^* of Player II in the one-step game $\mathbb{G}(V^{\mathbb{G},\ell})$. We may see τ^* as a memoryless strategy of Player II in the generalized reachability game $\mathbb{G}(\mathcal{R}^{\mathbb{G},\ell})$. We show that τ^* satisfies the inequality $\sup_{\sigma\in\Sigma_{\mathrm{I}}} P_s^{\sigma,\tau^*}(\mathcal{R}^{\mathbb{G},\ell}) \leq V^{\mathbb{G},\ell}(s)$ for any $s \in S$. (Thus, if we prove the second inequality, then we can say this τ^* is, in fact, an optimal strategy of Player II in the game $\mathbb{G}(\mathcal{R}^{\mathbb{G},\ell})$.) It is enough to show that $\sup_\sigma P_s^{\sigma,\tau^*}(\mathcal{R}_n^{\mathbb{G},\ell}) \leq V^{\mathbb{G},\ell}(s)$ for any $s \in S$ and $n \in \mathbb{N}$. We show this by induction on n. The case $n = 0$ is clear. Suppose that $\sup_\sigma P_s^{\sigma,\tau^*}(\mathcal{R}_n^{\mathbb{G},\ell}) \leq V^{\mathbb{G},\ell}(s)$ holds for any $s \in S$ as an induction hypothesis. Fix $s \in S$. If $s \in \mathrm{dom}(\ell)$, then it is obvious that the inequality holds for s. Otherwise, we have the equality $P_s^{\sigma,\tau^*}(\mathcal{R}_{n+1}^{\mathbb{G},\ell}) = \sum_{s'\in S} P_s^{\sigma,\tau^*}([ss'])P_{s'}^{\sigma,\tau^*}(\mathcal{R}_n^{\mathbb{G},\ell})$ for any $\sigma \in \Sigma_{\mathrm{I}}$. By the induction hypothesis, we know that $P_s^{\sigma,\tau^*}(\mathcal{R}_{n+1}^{\mathbb{G},\ell}) \leq \sum_{s'\in S} P_s^{\sigma,\tau^*}([ss'])V^{\mathbb{G},\ell}(s')$. Hence the equalities

$$\sup_\sigma P_s^{\sigma,\tau^*}(\mathcal{R}_{n+1}^{\mathbb{G},\ell}) \leq \sup_\sigma \sum_{s'\in S} P_s^{\sigma,\tau^*}([ss'])V^{\mathbb{G},\ell}(s') = V^{\mathbb{G},\ell}(s)$$

hold by the optimality of τ^* in the one-step game. Let us now show the second inequality. We have $P_s^{\sigma,\tau}(\mathcal{R}_n^{\mathbb{G},\ell}) \leq P_s^{\sigma,\tau}(\mathcal{R}^{\mathbb{G},\ell})$ since $\mathcal{R}_n^{\mathbb{G},\ell}(w) \leq \mathcal{R}^{\mathbb{G},\ell}(w)$ for any $w \in \Omega$. Hence $\sup_\sigma \inf_\tau P_s^{\sigma,\tau}(\mathcal{R}_n^{\mathbb{G},\ell}) \leq \sup_\sigma \inf_\tau P_s^{\sigma,\tau}(\mathcal{R}^{\mathbb{G},\ell})$ holds. By Theorem 3, $V_n^{\mathbb{G},\ell}(s) = \mathrm{val}_s(\mathcal{R}_n^{\mathbb{G},\ell}) = \sup_\sigma \inf_\tau P_s^{\sigma,\tau}(\mathcal{R}_n^{\mathbb{G},\ell})$ holds. Thus the second inequality holds. □

Corollary 1. *Player II has a memoryless and randomized optimal strategy in any generalized reachability game.* □

Contrary to the case of Player II, Player I has no even optimal strategy in some reachability games. We give such an example below.

Example 1. Consider the following simultaneous reachability game as shown in Figure 1. Let $S = \{s_0, s_1, s_2\}$, $A_{\mathrm{I}} = \{x_1, x_2\}$ and $A_{\mathrm{II}} = \{y_1, y_2\}$. Define a transition function δ by $\delta(s_0, x_1, y_1) = s_0$, $\delta(s_0, x_2, y_2) = s_2$, $\delta(s_0, x_1, y_2) = \delta(s_0, x_2, y_1) = s_1$ and $\delta(s_i, x, y) = s_i$ for any $i \in \{1, 2\}$ and $(x, y) \in A_{\mathrm{I}} \times A_{\mathrm{II}}$. Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$. Now consider the reachability game $\mathbb{G}(\mathcal{R}_{\{s_1\}})$.

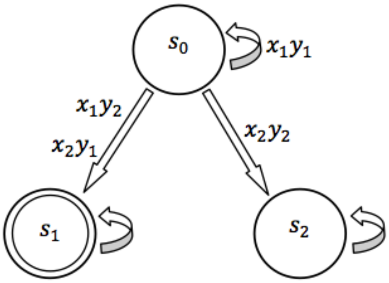


Fig. 1. An illustration of reachability game

One can prove that $\mathrm{val}_{s_0}(\mathcal{R}_{\{s_1\}}) = 1$. We show that Player I has no optimal strategy in the reachability game $\mathbb{G}(\mathcal{R}_{\{s_1\}})$.

Proof. Fix a strategy $\sigma \in \Sigma_{\mathrm{I}}$. We construct $\tau \in \Sigma_{\mathrm{II}}$ such that $P_{s_0}^{\sigma,\tau}(\mathcal{R}_{\{s_1\}}) < 1$. For $\rho \in \Omega^{\mathrm{fin}}(\mathbb{G})$, define $\tau(\rho)(y_1) = 1$ if $\sigma(\rho)(x_1) = 1$, and define $\tau(\rho)(y_2) = 1$ otherwise. It is clear that $P_{s_0}^{\sigma,\tau}(\mathcal{R}_{\{s_1\}}) < 1$ by the definitions of $\mathbb{G}$ and τ. □

The next theorem says that, given a generalized reachability game, Player I always has a memoryless ε-optimal strategy in this game for any positive real number ε. In fact, this result for reachability games was shown by Chatterjee et al. [5] in a slightly different setting. We essentially use their method to prove our theorem.

Theorem 5. *In every generalized reachability game $\mathbb{G}(\mathcal{R}^{\mathbb{G},\ell})$, there exist an ε-optimal memoryless strategy of Player I for any $\varepsilon > 0$.*

Proof. Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game and let ℓ a label on S. Without loss of generality, we may assume that if $s \in \mathrm{dom}(\ell)$ or $\mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell}) = 0$, then $\delta(s, x, y) = s$ holds for any $(x, y) \in A_{\mathrm{I}} \times A_{\mathrm{II}}$.

Fix a positive real $\varepsilon > 0$. Choose $n \in \mathbb{N}$ such that for any $s \in S$, the inequality $V_{n-1}^{\mathbb{G},\ell}(s) \geq \mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell}) - \varepsilon$ holds, and $\mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell}) > 0$ implies $V_{n-1}^{\mathbb{G},\ell}(s) > 0$. For $m \leq n$, choose $\sigma_m \in \Sigma_{\mathrm{I}}^{\mathrm{M}}$ such that σ_m is an optimal strategy of Player I in the one-step game $\mathbb{G}(V_{m-1}^{\mathbb{G},\ell})$. We define a strategy $\sigma^* \in \Sigma_{\mathrm{I}}^{\mathrm{M}}$ by $\sigma^*(s) = \sigma_{m_s}(s)$ for any $s \in S$, where m_s is the least number $m \leq n$ such that $V_m^{\mathbb{G},\ell}(s) = V_n^{\mathbb{G},\ell}(s)$. By the definition, $V_{m_s}^{\mathbb{G},\ell}(s) = \inf_{\tau \in \Sigma_{\mathrm{II}}^{\mathrm{M}}} P_s^{\sigma^*,\tau}(V_{m_s-1}^{\mathbb{G},\ell})$ holds for any $s \in S \setminus \mathrm{dom}(\ell)$. Now choose a strategy $\tau^* \in \Sigma_{\mathrm{II}}^{\mathrm{M}}$ such that $P_s^{\sigma^*,\tau^*}(\mathcal{R}^{\mathbb{G},\ell}) = \inf_\tau P_s^{\sigma^*,\tau}(\mathcal{R}^{\mathbb{G},\ell})$ for all $s \in S$.

Fix a $s \in S \setminus \mathrm{dom}(\ell)$ with $V_{m_s}^{\mathbb{G},\ell}(s) > 0$. Suppose that $V_n(s) \geq V_n(s')$ holds for any $s' \in S$ with $P_s^{\sigma^*,\tau^*}([ss']) > 0$. We have $V_{m_s}^{\mathbb{G},\ell}(s) = V_{m_s-1}^{\mathbb{G},\ell}(s') = V_n^{\mathbb{G},\ell}(s')$ for any $s' \in S$ with $P_s^{\sigma^*,\tau^*}([ss']) > 0$ since $V_n^{\mathbb{G},\ell}(s) = V_{m_s}^{\mathbb{G},\ell}(s)$, $V_{m_s-1}^{\mathbb{G},\ell}(s') \leq V_n^{\mathbb{G},\ell}(s')$ and $V_{m_s}^{\mathbb{G},\ell}(s) \leq P_s^{\sigma^*,\tau}(V_{m_s-1}^{\mathbb{G},\ell})$ hold. Therefore, if $s' \in S$ satisfies $P_s^{\sigma^*,\tau^*}([ss'])$, then $m_s > m_{s-1}$. As a result, we know that for any $s \in S \setminus \mathrm{dom}(\ell)$ there exists s' with $P_s^{\sigma^*,\tau^*}([ss']) > 0$ such that

$$V_n^{\mathbb{G},\ell}(s) < V_n^{\mathbb{G},\ell}(s') \text{ or } m_s > m_{s-1}.$$

Note that $\{V_n^{\mathbb{G},\ell}(s) : s \in S \setminus \mathrm{dom}(\ell)\}$ is finite, and $m_s = 0$ implies $s \in \mathrm{dom}(\ell)$ or $V_n^{\mathbb{G},\ell}(s) = 0$. Here $V_n^{\mathbb{G},\ell}(s) = 0$ implies $\mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell}) = 0$. Hence for any $s \in S$ there exists $\rho \in \Omega_s^{\mathrm{fin}}$ such that $P_s^{\sigma^*,\tau^*}([\rho]) > 0$ and

$$\rho(|\rho|-1) \in \mathrm{dom}(\ell) \text{ or } \mathrm{val}_{\rho(|\rho|-1)}(\mathcal{R}^{\mathbb{G},\ell}) = 0.$$

As a conclusion, we have $P_s^{\sigma^*,\tau^*}(A) = 0$ for any $s \in S$, where $A = \{w \in \Omega : (\forall n \in \mathbb{N})[w(n) \in \mathrm{dom}(\ell) \ \&\ \mathrm{val}_{w(n)}(\mathcal{R}^{\mathbb{G},\ell}) > 0]\}$. Thus, the sum

$$\sum \left\{ V_n^{\mathbb{G},\ell}(\rho(|\rho|-1)) P_s^{\sigma^*,\tau^*}([\rho]) : \rho \in \Omega_s^{\mathrm{fin}} \ \&\ |\rho| = k \right\}$$

tends to $P_s^{\sigma^*,\tau^*}(\mathcal{R}^{\mathbb{G},\ell})$ as k to ∞. It is easy to see by induction on $k \in \mathbb{N}$ that

$$\sum \left\{ V_n^{\mathbb{G},\ell}(\rho(|\rho|-1)) P_s^{\sigma^*,\tau^*}([\rho]) : \rho \in \Omega_s^{\mathrm{fin}} \ \&\ |\rho| = k \right\} \geq V_{n-1}^{\mathbb{G},\ell}(s)$$

holds for any $k \in \mathbb{N}$. Hence we have $P_s^{\sigma^*,\tau^*}(\mathcal{R}^{\mathbb{G},\ell}) \geq V_{n-1}^{\mathbb{G},\ell}(s) \geq \mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell}) - \varepsilon$. □

4 Büchi Games

Our plan in this section is as follows. After giving the definition of Büchi games, we describe values of Büchi games as values of some generalized reachability games. The proof includes the information how we can construct ε-optimal strategies in a Büchi game for a given positive real number ε. But we see that, in general, Player I may not have a memoryless ε-optimal strategy in a Büchi game for some positive real ε.

Definition 9. *Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game. For a run $w \in \Omega(\mathbb{G})$, we define* $\mathrm{Inf}(w)$ *as the set $\{s \in S : (\forall n \in \mathbb{N})(\exists m \geq n)[w(m) = s]\}$. For $T \subset S$, we set $\mathcal{B}^{\mathbb{G},T} = \{w \in \Omega(\mathbb{G}) : \mathrm{Inf}(w) \cap T \neq \emptyset\}$. Any game of the form $\mathbb{G}(\mathcal{B}^{\mathbb{G},T})$ is called a Büchi game. T is called the set of target states of the Büchi game $\mathbb{G}(\mathcal{B}^{\mathbb{G},T})$.*

In [7], they introduced quantitative game μ-calculus, and showed that the maximal probability of winning for Büchi game can be expressed as a fixpoint formulas. In particular, they characterized the optimality and the memory requirements of the winning strategies, that is, memoryless strategies suffice for winning games with reachability condition, and Büchi conditions require the use of strategies with infinite memory.

For a label ℓ on S, define $\mathcal{R}_+^{\mathbb{G},\ell} : \Omega(\mathbb{G}) \to [0,1]$ by

$$\mathcal{R}_+^{\mathbb{G},\ell}(w) = \begin{cases} \ell(w(N_w)) & \text{if } (\exists N > 0)[w(N) \in \operatorname{dom}(\ell)], \\ 0 & \text{otherwise,} \end{cases}$$

where N_w is the least natural number $N > 0$ such that $w(N) \in \operatorname{dom}(\ell)$. Clearly, the results for generalized reachability games in the previous section hold even for games of the form $\mathbb{G}(\mathcal{R}_+^{\mathbb{G},\ell})$. We also call this kind of games generalized reachability games. To study Büchi games, these results for games of the form $\mathbb{G}(\mathcal{R}_+^{\mathbb{G},\ell})$ are useful.

Definition 10. *Let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game and let $T \subset S$. For any $n \in \mathbb{N}$, we define a label $\ell_n^{\mathbb{G},T}$ on S with the domain T inductively by*

$$\ell_0^{\mathbb{G},T}(s) = 1 \qquad\qquad \ell_{n+1}^{\mathbb{G},T}(s) = \operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell_n^{\mathbb{G},T}})$$

We set $\ell^{\mathbb{G},T}(s) = \lim_{n\to\infty} \ell_n^{\mathbb{G},T}(s)$.

For $T \subset S$ and $n \in \mathbb{N}$, we define $\mathcal{B}_n^{\mathbb{G},T} = \{w \in \Omega(\mathbb{G}) : (\exists^{\geq n} k > 0)[w(k) \in T]\}$. Here, read "$\exists^{\geq n} k > 0$" as "there exist at least n-many natural numbers $k > 0$".

Theorem 6. *The equality $\operatorname{val}_s^{\mathbb{G}}(\mathcal{B}_{n+1}^{\mathbb{G},T}) = \operatorname{val}_s^{\mathbb{G}}(\mathcal{R}_+^{\mathbb{G},\ell_n^{\mathbb{G},T}})$ holds for any $n \in \mathbb{N}$ and any $s \in S$.*

Proof. Let $\ell_n = \ell_n^{\mathbb{G},T}$ for any $n \in \mathbb{N}$. We show the equality holds by induction on n. Fix n. It is clear that the equation holds for any $s \in S$ when $n = 0$. Now, suppose that $\operatorname{val}_s(\mathcal{B}_{n+1}^{\mathbb{G},T}) = \operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell_n})$ holds for any $s \in S$. Then we have $\ell_{n+1}(s) = \operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell_n}) = \operatorname{val}_s(\mathcal{B}_{n+1}^{\mathbb{G},T})$ for any $s \in T = \operatorname{dom}(\ell_{n+1})$. Therefore, $\operatorname{val}_s(\mathcal{B}_{n+2}^{\mathbb{G},T}) = \operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell_{n+1}})$ holds for any $s \in S$. □

Theorem 7. *The equality $\operatorname{val}_s^{\mathbb{G}}(\mathcal{B}^{\mathbb{G},T}) = \operatorname{val}_s^{\mathbb{G}}(\mathcal{R}_+^{\mathbb{G},\ell^{\mathbb{G},T}})$ holds for any $s \in S$.*

Proof. Let $\ell = \ell^{\mathbb{G},T}$ and $\ell_n = \ell_n^{\mathbb{G},T}$ for any $n \in \mathbb{N}$. It is enough to show that the following inequalities:

$$\inf_{\tau\in\Sigma_{\mathrm{II}}} \sup_{\sigma\in\Sigma_{\mathrm{I}}} P_s^{\sigma,\tau}(\mathcal{B}^{\mathbb{G},T}) \leq \operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) \leq \sup_{\sigma\in\Sigma_{\mathrm{I}}} \inf_{\tau\in\Sigma_{\mathrm{II}}} P_s^{\sigma,\tau}(\mathcal{B}^{\mathbb{G},T}).$$

Note that $\mathcal{B}_n^{\mathbb{G},T} \supset \mathcal{B}^{\mathbb{G},T}$ holds for any $n \in \mathbb{N}$. Thus

$$\inf_{\tau} \sup_{\sigma} P_s^{\sigma,\tau}(\mathcal{B}^{\mathbb{G},T}) \leq \operatorname{val}_s(\mathcal{B}_{n+1}^{\mathbb{G},T}) = \operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell_n})$$

holds for any $n \in \mathbb{N}$. Since the righthand tends to $\operatorname{val}_s(\mathcal{R}_+^{\mathbb{G},\ell})$ as n to ∞, we know that the first inequality holds. Let us now show the second inequality also holds.

Fix a positive real ε. We construct a strategy $\sigma^* \in \Sigma_{\mathrm{I}}$ such that $\mathrm{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) - \varepsilon \leq \inf_\tau P_s^{\sigma^*,\tau}(\mathcal{B}^{\mathbb{G},T})$ holds for any $s \in S$. Choose sequences $\{\alpha_n\}_{n\in\mathbb{N}}$, $\{\beta_n\}_{n\in\mathbb{N}}$ of positive reals such that

$$\mathrm{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) - \varepsilon \leq \mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell})) \prod_{n\in\mathbb{N}}(1-\alpha_n), \quad \mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell})(1-\alpha_n) \leq \mathrm{val}_s(\mathcal{R}^{\mathbb{G},\ell}) - \beta_n$$

holds for any $s \in S$ and $n \in \mathbb{N}$. Choose a sequence $\{\sigma_n\}_{n\in\mathbb{N}}$ of strategies of Player I in the game $\mathbb{G}(\mathcal{R}^{\mathbb{G},\ell})$ such that σ_n is β_n-optimal for any $n \in \mathbb{N}$. For $k \in \mathbb{N}$, we define $\sigma_k^* \in \Sigma_{\mathrm{I}}$ as follows: for $\rho \in \Omega^{\mathrm{fin}}$, let $\sigma_k^*(\rho) = \sigma_{n+k}(\rho_{\mathrm{suf}})$, where $\rho_{\mathrm{suf}} \in \Omega^{\mathrm{fin}}(\mathbb{G})$ satisfies $\rho = (\rho \upharpoonright m)\rho_{\mathrm{suf}}$ with $m = \max\{0, i-1 : \rho(i) \in T\}$, and $n = \#\{i \leq m : \rho(i) \in T\}$. We show by induction on n that the inequality

$$\mathrm{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) \prod_{i\leq n}(1-\alpha_{i+k}) \leq \inf_\tau P_s^{\sigma_k^*,\tau}(\mathcal{B}_n^{\mathbb{G},T}) \tag{1}$$

holds for any $s \in S$ and $k, n \in \mathbb{N}$. Fix $n \in \mathbb{N}$. Clearly (1) holds for any $s \in S$ and $k \in \mathbb{N}$ when $n = 0$. Now, suppose that n satisfies (1) for any $s \in S$ and $k \in \mathbb{N}$. For $s, s' \in S$, define $A_{s,s'}$ as the set $\{\rho \in \Omega_s^{\mathrm{fin}} : \rho(|\rho|-1) = s' \;\&\; (\forall i \in \{1, \cdots, |\rho|-2\})[\rho(i) \notin T]\}$. The inequalities

$$\begin{aligned}
&\inf_\tau P_s^{\sigma_k^*,\tau}(\mathcal{B}_{n+1}^{\mathbb{G},T}) \\
\geq &\inf_\tau \sum \left\{ P_s^{\sigma_k^*,\tau}([\rho]) \inf_{\tau'} P_{s'}^{\sigma_{k+1}^*,\tau'}(\mathcal{B}_n^{\mathbb{G},T}) : \rho \in A_{s,s'} \;\&\; s' \in T \right\} \\
\geq &\inf_\tau \sum \left\{ P_s^{\sigma_k^*,\tau}([\rho]) \mathrm{val}_{s'}(\mathcal{R}_+^{\mathbb{G},\ell}) \prod_{i\leq n}(1-\alpha_{i+k+1}) : \rho \in A_{s,s'} \;\&\; s' \in T \right\} \\
\geq &\mathrm{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) \prod_{i\leq n+1}(1-\alpha_{i+k})
\end{aligned}$$

holds for any $s \in S$ and $k \in \mathbb{N}$. Thus (1) holds for any $s \in S$ and $k, n \in \mathbb{N}$. Hence $\mathrm{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) - \varepsilon \leq \inf_\tau P_s^{\sigma_0^*,\tau}(\mathcal{B}_n^{\mathbb{G},T})$ holds for any $s \in S$ and $n \in \mathbb{N}$. Since $\mathcal{B}^{\mathbb{G},T} = \bigcap_{n\in\mathbb{N}} \mathcal{B}_n^{\mathbb{G},T}$ holds, we have $\mathrm{val}_s(\mathcal{R}_+^{\mathbb{G},\ell}) - \varepsilon \leq \inf_\tau P_s^{\sigma_0^*,\tau}(\mathcal{B}^{\mathbb{G},T})$ for any $s \in S$. □

The following example shows that there exist a positive real ε and a Büchi game in which Player I has no ε-optimal memoryless strategy.

Example 2. Let $\mathbb{G}(\mathcal{B}^{\mathbb{G},T})$ be the simultaneous Büchi game given as Figure 2.

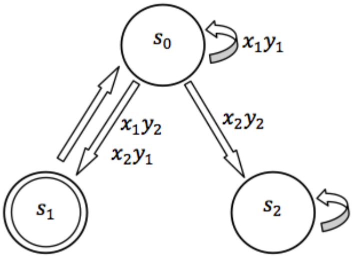


Fig. 2. An illustration of Büchi game

That is, let $\mathbb{G} = (S, A_{\mathrm{I}}, A_{\mathrm{II}}, \delta)$ be a game, where $S = \{s_0, s_1, s_2\}$, $A_{\mathrm{I}} = \{x_1, x_2\}$ and $A_{\mathrm{II}} = \{y_1, y_2\}$, and the transition function δ is given by $\delta(s_0, x_1, y_1) = s_0$, $\delta(s_0, x_2, y_2) = s_2$, $\delta(s_0, x_1, y_2) = \delta(s_0, x_2, y_1) = s_1$, $\delta(s_1, x, y) = s_0$ and $\delta(s_2, x, y) = s_2$ for all $(x, y) \in A_{\mathrm{I}} \times A_{\mathrm{II}}$. Define $T = \{s_1\}$. One can prove that $\mathrm{val}_{s_0}(\mathcal{B}^{\mathbb{G},T}) = 1$. We show that Player I has no ε-optimal memoryless strategy in the Büchi game $\mathbb{G}(\mathcal{B}^{\mathbb{G},T})$ for any positive real $\varepsilon < 1$.

Proof. Fix a positive real $\varepsilon < 1$ and a strategy $\sigma \in \Sigma_{\mathrm{I}}^{\mathrm{M}}$. We define a strategy $\tau \in \Sigma_{\mathrm{II}}$ by $\tau(\rho)(y_1) = 1$ if $\sigma(\rho)(x_1) = 1$, and $\tau(\rho)(y_2) = 1$ otherwise. In the case that $\sigma(s_0)(x_1) = 1$, we have $P_{s_0}^{\sigma,\tau}(\{s_0^{\mathbb{N}}\}) = 1$. Otherwise, we have $P_{s_0}^{\sigma,\tau}(\mathcal{R}^{\mathbb{G},\{s_2\}}) = 1$ since $\sigma(\rho) = \sigma(\rho')$ if $\rho(|\rho| - 1) = \rho'(|\rho'| - 1)$. Therefore, $\inf_{\tau'} P_{s_0}^{\sigma,\tau'}(\mathcal{B}^{\mathbb{G},\{s_1\}}) = 0 < 1 - \varepsilon = \mathrm{val}_{s_0}(\mathcal{B}^{\mathbb{G},T}) - \varepsilon$ holds. □

Acknowledgments. The first author would like to thank Prof. Kazuyuki Tanaka for encouragement and support. The second author acknowledges financial support from JSPS research fellowships.

References

[1] Blackwell, D.: Infinite G_δ games with imperfect information. In: Zastosowania Matematyki Applicationes Mathematicae, Hugo Steinhaus Jubilee Volume, pp. 99–101 (1969)

[2] Chatterjee, K., Randour, M., Raskin, J.-F.: Strategy synthesis for multi-dimensional quantitative objectives. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 115–131. Springer, Heidelberg (2012)

[3] Chatterjee, K., Henzinger, T.A., Jurdzinski, M.: Mean-payoff parity games. In: Proc. of LICS, pp. 178–187. IEEE Computer Society (2005)

[4] Chatterjee, K., Doyen, L., Singh, R.: On memoryless quantitative objectives. In: Owe, O., Steffen, M., Telle, J.A. (eds.) FCT 2011. LNCS, vol. 6914, pp. 148–159. Springer, Heidelberg (2011)

[5] Chatterjee, K., de Alfaro, L., Henzinger, T.A.: Strategy improvement for concurrent reachability games. In: Proc. of Third International Conference on the Quantitative Evaluation of Systems QEST (2006)

[6] De Alfaro, L., Henzinger, T.A., Majumdar, R.: Discounting the future in systems theory. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 1022–1037. Springer, Heidelberg (2003)

[7] De Alfaro, L., Majumdar, R.: Quantitative solution of omega-regular games. Journal of Computer and System Sciences 68, 374–397 (2004)

[8] Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. Journal of Game Theory 8(2), 109–113 (1979)

[9] Filar, J., Vrieze, K.: Competitive Markov decision processes. Springer, Heidelberg (1997)

[10] Gale, D., Stewart, F.M.: Infinite games with perfect information. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games. Annals of Mathematics Studies, vol. 2(28), pp. 245–266. Princeton University Press, Princeton (1953)

[11] Krcăl, J.: Determinacy and optimal strategies in stochastic games. Master's Thesis, Faculty of Informatics, Masaryk University (2009)

[12] Kučera, A.: Turn-based stochastic games. In: Apt, K.R., Grädel, E. (eds.) Lectures in Game Theory for Computer Scientists, pp. 146–184. Cambridge University Press, Cambridge (2011)

[13] Martin, D.A.: The determinacy of Blackwell games. Journal Symbolic Logic 63(4), 1565–1581 (1998)

[14] Martin, D.A.: Borel determinacy. Annals of Mathematics 102, 363–371 (1975)

[15] Von Neumann, J.: Zur Theorie der Gesellschaftsspiele. Math. Annalen 100, 295–320 (1928)

[16] Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. Theoretical Computer Science 158, 343–359 (1996)

Berge-Zhukovskii Optimal Nash Equilibria

Noémi Gaskó, Mihai Suciu, Rodica Ioana Lung, and Dan Dumitrescu

Babes-Bolyai University of Cluj-Napoca, Romania

Abstract. The Berge-Zhukovskii optimal Nash equilibrium combines the properties of the popular Nash equilibrium with the ones of the less known Berge-Zhukovskii by proposing yet another Nash equilibrium refinement. Moreover, a computational approach for the detection of these newly proposed equilibria is presented with examples for two auction games.

1 Introduction

The most used equilibrium concept in non-cooperative game theory is the Nash equilibrium [16], which assumes rational players that care only about themselves and make rational choices to achieve the best possible payoffs. There are many criticisms brought to Nash equilibria: it does not ensure the highest payoff for players (e.g. trust games, social dilemmas); in some cases it may not exist in pure form and in other cases a game may present an infinite number of equilibria [11].

One way to solve the problem of multiple equilibria is to consider some refinements of the Nash equilibrium (e.g. strong Nash [2], coalition proof Nash [4], etc.). Another way to approach the problem is to propose alternatives, such as the Berge-Zhukovskii (BZ) equilibrium [21]. While Nash equilibrium is stable against unilateral deviations of players, BZ is stable against deviations of all other players. Berge-Zhukovskii equilibrium can be interpreted as an other-regarding, altruistic equilibrium [5].

The Berge-Zhukovskii optimal Nash (BZON) equilibrium as a new refinement for Nash equilibrium is introduced that presents BZ properties, which are characterized by the use of a generative relation.

A Differential evolution (DE) [6] algorithm based on the aforementioned generative relation is used to compute BZON equilibria for first-price and second-price auction games illustrating the approach.

The article is organized as follows: the next section presents some basic game theory notions and concepts. The DE is described in Section 3. The fourth section presents numerical experiments. The paper ends with Conclusions.

2 Game Equilibria

A finite strategic non-cooperative game is defined by the set of players involved in the game, a set of possible actions associated with each player and their corresponding payoffs. Formally, a game is a system $G = ((N, S_i, u_i), i = 1, ..., n)$, where:

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_4, © Springer International Publishing Switzerland 2014

- N represents the set of players, n is the number of players;
- S_i is the set of actions available to player $i \in N$, and $S = S_1 \times S_2 \times ... \times S_n$ is the set of all possible strategies of the game, $s = (s_1, ...s_n) \in S$ is a strategy (or strategy profile) of the game;
- for each player $i \in N$, $u_i : S \to R$ represents the payoff function for player i.

The most popular and used equilibrium concept is the Nash equilibrium [16]. When playing in Nash sense no player can improve its payoff by deviating from its strategy only by himself.

Let us denote by (s_i, s^*_{-i}) the strategy profile obtained from s^* by replacing the strategy of player i with $s_i : (s_i, s^*_{-i}) = (s^*_1, ..., s_i, ..., s^*_n)$.

Definition 1 (Nash equilibrium). *A strategy profile $s^* \in S$ is a Nash equilibrium if*

$$u_i(s_i, s^*_{-i}) \leq u_i(s^*),$$

holds $\forall i = 1, ..., n, \forall s_i \in S_i$.

We will denote by

$$S_{-i} = S_1 \times ... \times S_{i-1} \times S_{i+1} \times ... \times S_n,$$

with

$$s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n)$$

and

$$(s^*_i, s_{-i}) = (s_1, s_2, ..., s^*_i, ..., s_n).$$

A more general equilibrium concept is the Berge equilibrium [3].

Definition 2 (Berge equilibrium). *Let M be a finite set of indices. Denote by $P = \{P_t\}, t \in M$ a partition of N and $R = \{R_t\}, t \in M$ be a set of subsets of N. A strategy profile $s^* \in S$ is an equilibrium strategy for the partition P with respect to the set R, or simply a Berge equilibrium strategy, if and only if the condition*

$$u_{p_m}(s^*) \geq u_{p_m}(s_{R_m}, s^*_{N-R_m})$$

holds for each given $m \in M$, any $p_m \in P_m$ and $s_{R_m} \in S_{R_m}$.

If we consider that each class P_i of partition P consists of one player i and each set R_i is the set of players N except i, we obtain the Berge-Zhukovskii equilibrium [21]. We have $M = N$, $P_i = \{i\}$ and $R_i = N - i, \forall i \in N,$.

Playing in Berge-Zhukovskii sense can be interpreted as each player maximizing the payoff of the other players. More formally:

Definition 3 (Berge-Zhukovskii equilibrium). *A strategy profile $s^* \in S$ is a Berge-Zhukovskii equilibrium if the inequality*

$$u_i(s^*) \geq u_i(s^*_i, s_{-i})$$

holds for each player $i = 1, ..., n$, and all $s_{-i} \in S_{-i}$.

The strategy s^* is a Berge-Zhukovskii equilibrium, if the payoff of each player i does not decrease considering any deviation of the other $N - \{i\}$ players.

Let $Q \subset S$ and $s^* \in S$ a strategy profile. Then s^* is a *Berge-Zhukovskii equilibrium with respect to* Q if the inequality

$$u_i(s^*) \geq u_i(s_i^*, q_{-i})$$

holds for each player $i = 1, ..., n$, and all $q_{-i} \in Q_{-i}$.

Definition 4 (Berge-Zhukovskii Optimal Nash). *A Nash equilibrium profile that is also a Berge-Zhukovskii with respect to the entire set of Nash equilibria is called a Berge-Zhukovskii optimal Nash equilibrium of the game.*

A Berge-Nash equilibrium is introduced in [1] as a Berge equilibrium which is also a Nash equilibrium.

Figure 1 illustrates the connection between all these equilibria.

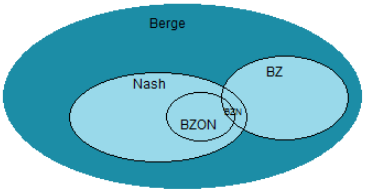


Fig. 1. Connection between Berge, Berge-Zhukovskii, Nash and BZON equilibria

3 BZON Detection

Due to all the similarities between multiobjective optimization problems [8] and non-cooperative games - the most important one being that both of them aim to optimize several payoff/objective functions in the same time - it is natural to assume that multiobjective optimization algorithms can also be used for game solving. Pareto based evolutionary approaches particularly can be suitable as they rely on the Pareto domination relation which can be changed with other relations redirecting the search towards different types of solutions/equilibria. However, the challenge is to find the appropriate relation for each equilibrium type.

Such a relation has been defined for Nash equilibria in [14] by using a quality measure $k(s, q)$ denoting the number of players that benefit from unilaterally switching their choices from s to q:

$$k(s, q) = card\{i \in N, u_i(q_i, s_{-i}) \geq u_i(s), q_i \neq s_i\},$$

where $card\{M\}$ denotes the cardinality of the set M.

Definition 5. *Let $q, s \in S$. We say the strategy q is better than strategy s with respect to Nash equilibrium (q Nash ascends s, and we write $q \prec_N s$, if the following inequality holds:*

$$k(q, s) < k(s, q).$$

Definition 6. *The strategy profile $q \in S$ is called Nash non-dominated, if and only if there is no strategy $s \in S, s \neq q$ such that*

$$s \prec_N q.$$

The relation $\prec_N$ is a generative relation for Nash equilibrium in the sense that the set of non-dominated strategies with respect to $\prec_N$ is equal to the set of Nash equilibria [14].

A similar quality measure exists also for the Berge-Zhukovskii equilibrium [9], counting how many players would benefit when all the others change their strategies from s to q :

$$b(s, q) = card\{i \in N, u_i(s) < u_i(s_i, q_{-i}), s_{-i} \neq q_{-i}\},$$

Definition 7. *Let $s, q \in S$. We say the strategy s is better than strategy q with respect to Berge-Zhukovskii equilibrium (s BZ-dominates q), and we write $s \prec_B q$, if and only if the inequality*

$$b(s, q) < b(q, s)$$

holds.

Definition 8. *The strategy profile $s^* \in S$ is a Berge-Zhukovskii non-dominated strategy (BZ nondominated), if and only if there is no strategy $s \in S, s \neq s^*$ such that s dominates s^* with respect to $\prec_B$ i.e.*

$$\nexists s \in S : s \prec_B s^*.$$

Relation $\prec_B$ is a generative relation of the Berge-Zhukovskii equilibrium.

The BZON equilibria can be computed by combining the above mentioned relations in the following manner:

1. The Nash ascendancy relation is checked first.
2. The BZ domination second.

Thus a new generative relation for BZON is obtained.

Crowding Based Differential Evolution. The Crowding based Differential evolutionary algorithm (CrDE) [20] can be easily adapted to compute different types of equilibria for static and dynamic games [10], [15], [19]. In a similar manner it can be used to compute BZON equilibria, by simply using the proposed generative relation when comparing two individuals.

4 Case Study: Auction Games

4.1 Prerequisites

Auction theory is important in economic transactions where an auction is a well defined micro-economic environment (for a survey see [12] or [18]).

The four main categories of auctions with complete information [7] are:

- first-price sealed bid auction - each bidder submits her/his own bid without seeing others bids, and the object is sold to the highest bidder at her/his bid;
- second-price sealed bid auction (Vickrey auctions)- each bidder submits her/his own bid, the object is sold to the highest bidder, who needs to pay only the second highest price for the object;
- open ascending-bid auctions (English auctions) - each bidder offers increasingly higher bids, the auction stops when no bidder wants to make a higher bid. The winner with the highest bid wins the object and needs to pay her/his own bid;
- open descending-bid auctions (Dutch auctions) - can be considered the inverse of the English auction, the initial price of the object is set by the auctioneer, the bidders lower the price, until there is no new bid. The winner pay her/his own bid.

Besides the mentioned categories other auction types are: all-pay auctions, Amsterdam auctions, unique bid-auctions, etc.

4.2 Game Theoretic Model of Auctions

Auction theory can be approached from different views: from a game theoretical perspective [13], from a market microstructure view, etc.

From a game theoretical view the auction has the following elements:

- players - the n bidders, $n \geq 2$;
- actions - the set of possible bids (b_i for the ith player);
- payoff function - depending on the type of auction, the player with the maximum bid gains the difference between the value of the object and maximum bid (or the difference between the value and the second highest bid - in second-price sealed bid auction);

We analyze some class of auctions from a game theoretical perspective. The first-price and second-price sealed bid auctions have several Nash equilibria. The aim is to show that it is possible to evolutionary compute the BZON equilibria.

4.3 Numerical Experiments

CrDE was run by using parameters presented in Table 1. For each experiment ten different runs were conducted.

Table 1. Parameter settings for CrDE used for the numerical experiments

Parameter	
Population size	50
Max no evaluations	2×10^7
CF	50
F	0.5
Crossover rate	0.9

First-Price Sealed Bid Auction. In the first-price sealed bid auction two players cast their bid independently. The value of the bidding objects is v_1 for the first player and v_2 for the second one. The winner is the highest bidder, who needs to pay his own bid (b_i). We can specify a simple agreement: if both have equal bids, the winner is the first bidder (another variant is to randomly choose a winner).

The payoff functions are the following [17]:

$$u_1(b_1, b_2) = \begin{cases} v_1 - b_1, & \text{if } b_1 \geq b_2, \\ 0, & \text{otherwise.} \end{cases}$$

$$u_2(b_1, b_2) = \begin{cases} v_2 - b_2, & \text{if } b_2 > b_1, \\ 0, & \text{otherwise.} \end{cases}$$

The game has several Nash equilibria as any $v_1 \leq b_1^* = b_2^* \leq v_2$ is a Nash equilibrium of the game. The BZON equilibrium of the game is a single strategy profile $(b_1, b_2) = (v_1, v_1)$.

For numerical experiments we consider the object values $v_1 = 5$, $v_2 = 3$, naturally this means that the maximal bid is less than 5 (nobody will cast more than the value of the object).

Numerical experiments over ten independent runs detect correctly the strategy profile $(b_1, b_2) = (v_2, v_2) = (3, 3)$. (with standard deviation 0.0).

Figure 2 illustrates the strategies space by using 50000 randomly generated strategies; the Nash equilibrium, Berge-Zhukovskii equilibrium and the Berge-Zhukovskii optimal Nash equilibrium of the game are presented. In Figure 3 the corresponding payoffs are depicted.

Let us consider the n-person version of the game, the payoff functions can be described as follows:

$$u_i(b_1, ..., b_n) = \begin{cases} v_i - b_i, & \text{if } b_i = max\{b_1, b_2, ..., b_n\} \\ 0, & \text{otherwise.} \end{cases}$$

For the three player version of the game with the object values $v_1 = 5$, $v_2 = 4$, $v_3 = 3$ we also have infinite set of Nash equilibria but only one (correctly detected) BZON equilibria $(b_1, b_2, b_3) = (4, 4, 0)$. It is an advantage that the bidder does not need to play the entire value of the bidding object.

Second-Price Sealed Bid Auction. In a second-price auction game the winner needs to pay the second highest bid, consider the two player version of the game [17]:

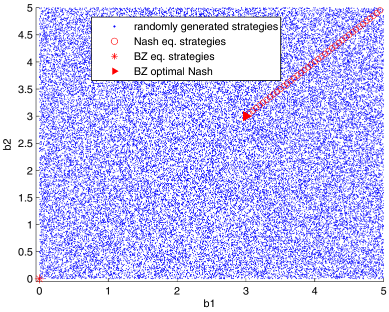


Fig. 2. Randomly generated strategies, Nash equilibria, Berge-Zhukovskii and Berge-Zhukovskii optimal Nash equilibrium strategies

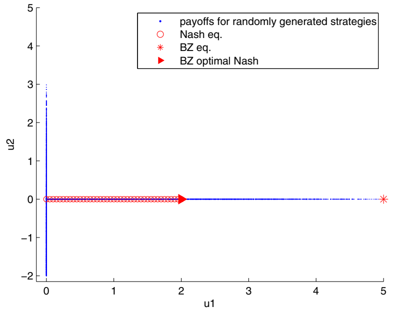


Fig. 3. Corresponding payoffs for randomly generated strategies, Nash equilibria, Berge-Zhukovskii and Berge-Zhukovskii optimal Nash equilibrium payoffs

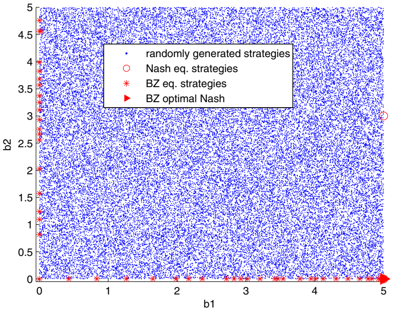


Fig. 4. Randomly generated strategies, Nash equilibria, Berge-Zhukovskii and Berge-Zhukovskii optimal Nash equilibrium strategies

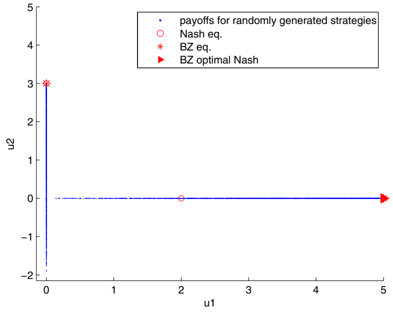


Fig. 5. Corresponding payoffs for randomly generated strategies, Nash equilibria, Berge-Zhukovskii and Berge-Zhukovskii optimal Nash equilibrium payoffs

$$u_1(b_1,b_2) = \begin{cases} v_1 - b_2, & \text{if } b_1 \geq b_2, \\ 0, & \text{otherwise.} \end{cases}$$

$$u_2(b_1,b_2) = \begin{cases} v_2 - b_1, & \text{if } b_2 > b_1, \\ 0, & \text{otherwise.} \end{cases}$$

In the second-price sealed bid auction also exist multiple Nash equilibria [17] every strategy profile $(b_1, b_2) = (v_1, v_2)$ is a Nash equilibrium of the game. Another equilibrium is $(b_1, b_2) = (v_1, 0)$, or $(b_1, b_2) = (v_2, v_1)$.

Let us consider $v_1 = 5$ and $v_2 = 3$. CrDE detects the strategy profile $(b_1, b_2) = (5, 0)$, also a single equilibrium point, which is the best possible outcome for the first player.

In Figure 4 the strategies space by using 50000 randomly generated strategies is illustrated; the Nash equilibrium, Berge-Zhukovskii equilibrium and the Berge-Zhukovskii optimal Nash equilibrium of the game are presented. In Figure 5 the corresponding payoffs are depicted.

5 Conclusion

This paper introduces the Berge-Zhukovskii optimal Nash equilibria as a refinment of Nash equilibria that are Berge-Zhukovskii with respect to the set of Nash equilibria. As Nash equilibria are stable against unilateral deviations and BZ are stable against the deviations of the others, the subset of Nash equilibria that present BZ properties among NEs is of interest both to rational and altruistic players.

Furthermore, the paper presents a simple way of computing the BZON by using a differential evolution algorithm. Numerical examples using auction games illustrate this approach.

Acknowledgments. The authors wish to thank the support of the OPEN-RES Academic Writing project 212/2012.

References

[1] Abalo, K., Kostreva, M.: Berge equilibrium: some recent results from fixed-point theorems. Applied Mathematics and Computation 169, 624–638 (2005)

[2] Aumann, R.J.: Acceptable points in general cooperative n-person games. In: Luce, R.D., Tucker, A.W. (eds.) Contribution to the Theory of Game IV. Annals of Mathematical Study, vol. 40, pp. 287–324. University Press (1959)

[3] Berge, C.: Théorie générale des jeux? n personnes. Gauthier-Villars (1957)

[4] Douglas Bernheim, B., Peleg, B., Whinston, M.D.: Coalition-proof Nash Equilibria I. Concepts. Journal of Economic Theory 42(1), 1–12 (1987)

[5] Colman, A.M., Korner, T.W., Musy, O., Tazdait, T.: Mutual support in games: Some properties of Berge equilibria. Journal of Mathematical Psychology 55(2), 166–175 (2011)

[6] Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. IEEE Trans. Evolutionary Computation 15(1), 4–31 (2011)

[7] Easley, D., Kleinberg, J.: Networks, crowds, and markets: Reasoning about a highly connected world (2012)

[8] Fonseca, C.M., Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms. ii. application example. IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans 28, 38–47 (1998)

[9] Gaskó, N., Dumitrescu, D., Lung, R.I.: Evolutionary Detection of Berge and Nash Equilibria. In: Pelta, D.A., Krasnogor, N., Dumitrescu, D., Chira, C., Lung, R. (eds.) NICSO 2011. SCI, vol. 387, pp. 149–158. Springer, Heidelberg (2011)

[10] Gaskó, N., Suciu, M.A., Lung, R.I., Dumitrescu, D.: Dynamic Generalized Berge-Zhukovskii equilibrium detection. In: Emmerich, M., et al. (eds.) A Bridge between Probability, Set Oriented Optimization, and Evolutionary Computation (EVOLVE), pp. 53–58 (2013)

[11] Halpern, J.Y.: Beyond nash equilibrium: Solution concepts for the 21st century. In: Proceedings of the Twenty-Seventh ACM Symposium on Principles of Distributed Computing, pp. 1–10. ACM (2008)

[12] Klemperer, P.: Auction theory: A guide to the literature. Journal of Economic Surveys 13(3), 227–286 (1999)

[13] Levin, J.: Auction theory (2004)

[14] Lung, R.I., Dumitrescu, D.: Computing nash equilibria by means of evolutionary computation. Int. J. of Computers, Communications & Control 6, 364–368 (2008)

[15] Lung, R.I., Dumitrescu, D.: A new evolutionary approach to minimax problems. In: 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 1902–1905. IEEE (2011)

[16] Nash, J.: Non-Cooperative Games. The Annals of Mathematics 54(2), 286–295 (1951)

[17] Osborne, M.J.: An Introduction to Game Theory. Oxford University Press (2004)

[18] Parsons, S., Rodriguez-Aguilar, J.A., Klein, M.: Auctions and bidding: A guide for computer scientists. ACM Comput. Surv. 43(2), 10:1–10:59 (2011)

[19] Suciu, M., Lung, R.I., Gaskó, N., Dumitrescu, D.: Differential evolution for discrete-time large dynamic games. In: 2013 IEEE Conference on Evolutionary Computation, June 20-23, vol. 1, pp. 2108–2113 (2013)

[20] Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: IEEE Proceedings of the Congress on Evolutionary Computation (CEC 2004), pp. 1382–1389 (2004)

[21] Zhukovskii, V.I., Chikrii, A.A.: Linear-quadratic differential games. In: Naukova Dumka, Kiev (1994)

A Generative Relation for Nash Equilibria on Symmetric Action Graph Games

Tudor Dan Mihoc

Babeş Bolyai University,
str. M. Kogalniceanu nr. 1,
Cluj Napoca
mihoct@cs.ubbluj.com

Abstract. A method to detect good approximations of Nash equilibria in Action Graph Games (games represented as graphs) based on evolutionary computation is presented in this paper. This technique can allow to represent, reason and solve symmetric games. A fitness solution based on non-domination for the search in the configuration set of an Action Graph Game is proposed.

Keywords: large games, action graph games, symmetric games, Nash equilibria.

1 Introduction

Since the proposal of mathematical games as tools for modelling conflicting situations, scholars searched for proper solutions – that have a meaning regarding the modelled reality – and also for methods to find those solutions. Nash equilibrium probably is the most famous one [5].

Games are regarded as units between a set of players, a set of actions available to each player and a set of payoff functions that each player aims to maximize. Over the years games have raised many challenges. Efficient computation of Nash equilibria remains, for example, after more than fifty years since the concept was proposed, one of the main unsolved aims in Computational Game Theory.

There were many attempts to find optimum methods of computing equilibria, each with its own advantages, but general methods to solve faster and reliable any game are still missing.

Another difficulty arises when the number of players increases. The problem is similar to the increase of objectives in multi-objective optimisation. On top of the computational difficulties related to the number of objective functions, large games are much more difficult to handle due the fact that are represented in normal form.

Action Graph Games (AGG) offer smaller representations [1]. If in the normal form a large game can reach an exponential representation, using AGG we will have a representation upper bounded by a polynomial function.

The paper is organized in four sections: first is the introduction followed by the prerequisites of Game Theory, where some basic notions are presented. In the third section the generative relation for Nash equilibria is extended on the configuration set. And the fourth part contains the conclusions.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_5, © Springer International Publishing Switzerland 2014

2 Prerequisites of Game Theory

In order to introduce the new method for equilibria detection some basic notions of Game Theory are presented.

2.1 Finite Strategic Games

We consider a **finite strategic game** defined as follows [4,7]:

Definition 1. *A finite strategic game is defined as* $\Gamma = ((N, S_i, u_i), i = \overline{1, n})$ *where:*

- $N = \{1, ..., n\}$ *the set of players,* n *is the number of players;*
- *for each player* $i \in N$*,* S_i *represents the set of actions available to him,* $S_i = \{s_{i1}, s_{i2}, ..., s_{im_i}\}$ *where* m_i *represents the number of strategies available to player* i *and* $S = S_1 \times S_2 \times ... \times S_n$ *is the set of all possible situations of the game and* $(s_1, s_2, ..., s_n) \in S$ *is a pure strategy profile;*
- *for each player* $i \in N$*,* $u_i : S \to \mathbb{R}$ *represents the pay-off function.*

2.2 Action Graph Games

Following [1] we define an action graph game (AGG).

Definition 2. *An* ***Action Graph*** $G = (\mathcal{A}, E)$ *is a directed graph where:*

- $\mathcal{A}$ *is the set of nodes. Each node* $\sigma \in \mathcal{A}$ *represents an action, so* $\mathcal{A} = \bigcup_{i \in N} S_i$ *is the set of distinct actions.*
- E *is the set of directed edges, where self edges are allowed. We say that* σ' *is a neighbour of* σ *if there is an edge from* σ' *to* σ*, i.e.* $(\sigma', \sigma) \in E$*. We denote with* $v(\sigma)$ *the set of neighbours of* σ*, i.e.*

$$v(\sigma) \equiv \{\sigma' \in \mathcal{A} | (\sigma', \sigma) \in E\}$$

Definition 3. *Given an action graph* $G = (\mathcal{A}, E)$ *and a set of strategy profiles* S*, a* ***configuration*** c *is a tuple of* $|\mathcal{A}|$ *non-negative integers* $(c(\sigma))_{\sigma \in \mathcal{A}}$*, where* $c(\sigma)$ *is the number of players who chose action* $\sigma \in \mathcal{A}$*, and where there is some* $s \in S$ *that would give rise to* c*. The set of all configurations is denoted as* C*. Let* $\mathcal{C} : S \to C$ *be the function that maps a strategy profile* s *to the corresponding configuration* c*: if* $c = \mathcal{C}(s)$ *then* $c(\sigma) = | \{i \in N : s_i = \sigma\} |$ *for all* $\sigma \in \mathcal{A}$

Definition 4. *An* ***Action Graph Game (AGG)*** *is a tuple* (N, S, G, U) *where*

- N *is the set of players;*
- S *is the set of strategy profiles;*
- $G = (\mathcal{A}, E)$ *is an action graph, where* $\mathcal{A} = \bigcup_{i \in N} S_i$ *is the set of distinct strategies*
- $U = (u^\sigma)_{\sigma \in \mathcal{A}}$ *is a tuple of* $|\mathcal{A}|$ *functions, where each* $u^\sigma : C^{(\sigma)} \to$ ***R*** *is the utility function for action* σ*; i.e.* $u^\alpha(c^{(\sigma)})$ *is the utility of a player who chooses* σ*, when the configuration is* $c^{(\sigma)}$*.*

According to [1] AGG representation of the games is a more compact one that can be used directly, without conversion to its induced normal form. An algorithm to compute the expected payoffs for games in AGG form in polynomial time exploiting their context-specific independence, anonymity and additivity structure has been developed by Jiang in 2012.

2.3 Examples of AGG

Two very simple examples of games represented as action graph game are presented. For these two examples the normal representation and the action graph representation have the same complexity, however for larger games the AGG are much more compact [1].

Prisoniers' dilemma The symmetric PD (prisoniers' dilemma) is one of the most well known two players strategic game. The game is described by the payoff matrix from Table 1.

Table 1. The strategic form of a symmetric PD with two players (Row and Column); R,S,T, and P are the players payoffs after playing **C** or **D** strategy. In each pair the first value is the payoff for the Row player and the second for the Column player.

	C	**D**
C	R, R	S, T
D	T, S	P, P

The following chain of inequalities must be satisfied:

$$T > R > P > S$$

There are two players, Row and Column. Each has two possible moves, **C** ”cooperate” or **D** ”defect”. For each possible pair of moves, the payoffs to Row and Column (in that order) are listed in the appropriate cell. R is the ”reward” payoff that each player receives if both cooperate (see Table 1). P is the ”punishment” that each receives if both defect. T is the ”temptation” that each receives if he alone defects and S is the payoff that he receives if he alone cooperates.

Formally we have the game:

$$\begin{aligned}&\Gamma = (N, S, U),\\ &N = \{Row, Column\},\\ &S_{Row} = \{C, D\}, S_{Column} = \{C, D\},\\ &U = (U_{Row}, U_{Column}).\end{aligned}$$

The action graph for this game is presented in Figure 1.

The set $\mathcal{A} = \{C, D\}$. To play on such representation each player places her option in one vertice (possibly the same one both).

Rock-Paper-Scissors: For example if we consider the classic two player game Rock-Paper-Scissors with the payoffs depicted in Table 2 we gave the action graph from Figure 2.

The set $\mathcal{A} = \{R, P, S\}$. To play on such representation each player places her option in one vertice (possibly the same one both).

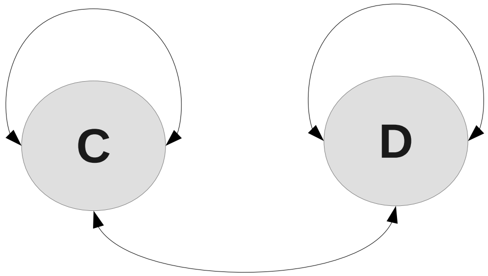


Fig. 1. The action graph for the Prisioners' Dilemma game

Table 2. The payoffs for the Rock-Paper-Scissors game

	R	P	S
R	0 , 0	-1 , 1	1, -1
P	1 , -1	0 , 0	-1 , 1
S	-1 , 1	1 , -1	0 , 0

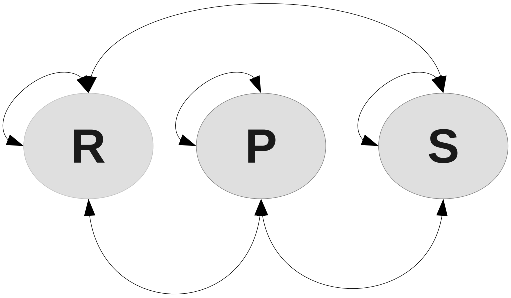


Fig. 2. The action graph for the Rock-Paper-Scissors game

3 Generative Relations

In this section the generative relation for Nash equilibria used for guiding an evolutionary algorithm will be presented. The generative relation will be extended in the configuration space in order to apply it directly on the AGG representation.

We will consider a action graph game G and we define the following binary relation.

3.1 Nash Ascendancy Relation for Strategies Profiles

Let us consider two pure strategy profiles s and s' from S. Let $k : S \times S \to \mathbb{N}$ be an operator that associates the cardinality of the set [2]

$$k(s, s') = |\{i \in \{1, ..., n\}|u_i(s'_i, s_{-i}) \geq u_i(s), s'_i \neq s_i\}|$$

to the pair (s, s').

This set is composed by the players i that would benefit if - given the strategy profile s - would change their strategy from s_i to s'_i, i.e.

$$u_i(s'_i, s_{-i}) \geq u_i(s).$$

Let $s, s' \in S$. The strategy profile s *Nash ascends* the strategy profile s' and we write $s \prec s'$ if the inequality

$$k(s, s') < k(s', s)$$

holds.

Thus a strategy profile s ascends strategy profile s' if there are less players that can increase their payoff by switching their strategy from s_i to s'_i than vice-versa.

Remark 1. Two strategy profiles $s, s' \in S$ may have the following relation:

1. either s dominates s', $s \prec s'$ $(k(s, s') < k(s', s))$
2. either s' dominates s, $s' \prec s$ $(k(s, s') > k(s', s))$
3. or $k(s, s') = k(s', s)$ and s and s' are considered indifferent (neither s dominates s' nor s' dominates s).

Thus this relation can be used as a relative measure of how ”close” of Nash is a pure strategy profile in comparison with another one [6].

3.2 Nash Ascendancy Relation in Configurations Space

In order to define a similar binary relation between configurations in an action-graph game we will consider a very important class of symmetric games, specifically the class where $S_i = \mathcal{A}, \forall i \in N$.

Let us consider two configurations c and c' from C.Let $\overline{k} : C \times C \to \mathbb{N}$ be an operator that associates the cardinality of the set

$$\begin{aligned} \overline{k}(c, c') = |\{(i, j) \in I_+ \times I_- | u_t(c, c_i - 1, c_j + 1) \geq u_t(c), I_+ = \{i \in \{1, ..., |\mathcal{A}|\} | \\ c_i - c'_i > 0\}, I_- = \{i \in \{1, ..., |\mathcal{A}|\} | c_i - c'_i < 0\}, \\ t - \text{the player that switches from strategy } i \text{ to } j\}| \quad (1) \end{aligned}$$

to the pair (c, c').

This set is composed of the pairs of strategies that players can change in order to cross from configuration c to configuration c', a change that would bring benefit to some player t, so the player would be tempted to change it's strategy.

Let $c, c' \in C$. The configuration c *Nash ascends* the configuration c' and we write $c \prec c'$ if the inequality

$$\overline{k}(c, c') < \overline{k}(c', c)$$

holds.

A configuration c ascends the configuration c' if there are less players that can increase their payoff by changing their strategy in order to switch from configuration c to c' than vice-versa.

Thus this relation can be used as a relative measure of how *close* of Nash is a configuration in comparison with another one.

Remark 2. We observe that for this special class of symmetric games the two relations are almost identical. Both of them count the number of players that can switch with success from one state of the game to the other, thus finding which one is closest to the Nash equilibria.

Remark 3. The domination relation in the set of configurations allows the detection of a hole class of strategy profiles, since usually a configuration coresponds more than one strategy profile. This is in accord with the symmetry of the game and of the Nash equilibria in such games.

4 Conclusions and Further Work

A new method for detecting Nash equilibria for symmetric games represented as action-graphs have been proposed. Instead of searching within the set of profile strategies (the states of the game) the search is conducted on the set of configurations of the game. Searching in this space allows one to reduce the number of evaluations.

The search uses a selection operator based on a domination relation between configurations, that can be used as a quality measure. Further work will imply adapting the method for non symmetric large games. Also the concept of a vicinity of a strategy within a configuration can be used to reduce the complexity of the computing process.

Acknowledgement. The authors would like to acknowledge also the support received from the Academic Writing project OPEN-RES PCCA1 212/2012.

References

[1] Jiang, A.X., Leyton-Brown, K., Bhat, N.: Action-Graph Games. Games and Economic Behavior 71(1), 141–173 (2011)

[2] Gaskó, N., Dumitrescu, D., Lung, R.I.: Evolutionary detection of berge and nash equilibria. In: Pelta, D.A., Krasnogor, N., Dumitrescu, D., Chira, C., Lung, R. (eds.) NICSO 2011. SCI, vol. 387, pp. 149–158. Springer, Heidelberg (2011)

[3] Lung, R.I., Dumitrescu, D.: Computing nash equilibria by means of evolutionary computation. Int. J. of Computers, Communications & Control III(suppl. issue), 364–368 (2008)

[4] McKelvey, R.D., McLennan, A.: Computation of equilibria in finite games. In: Handbook of Computational Economics, vol. 1, pp. 87–142 (1996)

[5] Nash, J.F.: Non-cooperative games. Annals of Mathematics 54, 286–295 (1951)

[6] Nagy, R., Suciu, M.A., Dumitrescu, D.: Lorenz equilibrium: equitability in non-cooperative games. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference. ACM (2012)

[7] Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)

Part III

Machine Learning Applied to Networks

Cognition: A Tool for Reinforcing Security in Software Defined Networks

Emilia Tantar, Maria Rita Palattella, Tigran Avanesov,
Miroslaw Kantor, and Thomas Engel

SnT, Université du Luxembourg,
4, rue Alphonse Weicker, L-2721 Luxembourg, Luxembourg
{emilia.tantar,maria-rita.palattella,tigran.avanesov,
miroslaw.kantor,thomas.engel}@uni.lu

Abstract. Security is one of the most important requirements for networks and serious concerns for network providers and users. Software Defined Networking offers to network managers new opportunities for deploying efficient security mechanisms. By means of applications and controller functionalities, SDN is able to provide a highly reactive network security monitoring, to perform comprehensive traffic analysis, and to enforce fine-grained dynamic access policies. In the present work, we show how such security mechanisms can be further enforced by applying cognitive functions at the SDN application plane. The proposed approach that finds its foundation into the control loops applied in Autonomic Managers Networks (AMNs), can efficiently enable secure and safe SDN scenarios.

Keywords: SDN, security, cognitive mechanisms.

1 Introduction

In the last years we have witnessed the progressive emergence of a new network paradigm, namely Software Defined Networking (SDN), that by introducing a centralized and programmable way of designing networks, has revolutionized the classical distributed networking approach. In detail, SDN separates the data plane (i.e., traffic forwarding between network devices, such as switches, routers, hosts, etc.) from the control plane (i.e., the decision making about the routing of traffic flows) and allows programmability of the network by external applications [44]. By means of a Southbound (SBI) and a Northbound (NBI) Interface, the control plane (i.e., the SDN controller) can interact, respectively, with the data plane and the application plane. Since the SDN's birth, the SBI has been standardized and commonly identified in the OpenFlow protocol [38], while the NBI is still under development.

SDN seems to be the promising network paradigm of choice for the future; and thus, the Open Networking Foundation (ONF), which has the leadership in SDN standardization, has recently put some effort for designing how to migrate from classical networks to SDN [43].

Born with the intention of overcoming the drawbacks of traditional networks (e.g., manual configuration and further maintenance of every single device in the network, latency in path-recovery due to the distributed approach, etc.), SDN has soon gained

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_6, © Springer International Publishing Switzerland 2014

the interest of companies, like Google, that have prefigured the advantages of applying it in their business networks [27].

SDN has quickly attracted the attention also of engineers and network designers in the standardization community (i.e., IETF, ITU-T), as shown by the newly ad hoc designed protocols for SDN, as the Forwarding and Control Element Separation Framework (ForCES) [65], and by the efforts conducted in the direction of extending and making classical protocols suitable for an SDN context. Ih this direction, it earths mentioning the Application-Layer Traffic Optimization Protocol (ALTO) [64], the Path Computation Element (PCE) Communication Protocol (PCEP) [62], the Bidirectional Forwarding Detection Protocol (BFD) [30] or the Extensible Messaging and Presence Protocol (XMPP) [53], to mention just a few.

Moreover, a new IEEE ComSoc Emerging Technology sub-committee focused on SDN and Network Functions Virtualization (NFV) has been recently created (December 2013), aiming to explore the aforementioned next generation networking technologies and their potential interaction with IPv6, cloud and mobility, which are fundamental IT design points [26].

Due to the new paradigm on which it is built, SDN has introduced several new challenges, in terms of performance, scalability, security and interoperability [55]. But, given (i) the importance for both end users and network providers to have a secure system, and (ii) the key-role played by security aspects in the maintenance and management of software defined networking, **security in SDN** has recently became one of the hottest arguments of discussion and investigation in the research community [31, 33, 48, 54, 57]. As pointed out in [41], the main aspects of interest from a security perspective are related to *authentication, identity management, monitoring, detection*, and *mitigation of threats*. To this aim, SDN controllers are enabled with a set of functions/modules that allow to monitor the current status of the whole network (e.g., network topology, switches' behavior, flows' path, etc.) and thus, to detect possible anomalies and thus, attacks from non-authorized malicious users.

In the present work, we have first outlined the main functions nowadays implemented in open-source SDN controllers, like OpenDayLight [45], FloodLight [18], POX [49], which in our humble opinion, can be efficiently exploited in monitoring, detection and (re)acting schemes for providing security in an SDN context. Our argument is supported by some related works (overviewed in the paper), that have already used some of these functionalities to develop SDN security solutions.

In the last decade, the increasing complexity of network infrastructures has raised up the conceptual need of building networks with autonomic management, see [36]. By following the same approach proposed for enabling Autonomic Network Management (ANM), i.e., by using intelligent mechanisms, such as bio-inspired techniques, or more generally particle methods in the application plane of an SDN architecture, we do believe it is possible to enhance the security that is achievable in a Software Defined Network, using only the controller functionalities. For instance, *classification techniques can play a role in building user profiles*, while other mechanisms (e.g., control loops, **cognitive algorithms**) *can be used in learning the various behavioral topologies and anticipating possible failures*, through rare events simulation.

The current work encloses an overview on how cognition enabled countermeasures can be applied for security failures in a centralized autonomous management perspective in a SDN scenario. Based on the advances available in literature, and with no pretension of being exhaustive, in the present work we have investigated the various facets and the intersection existing between autonomous networking, cognitive networks, and SDN, with a particular focus on security challenges. As compared with the autonomous networking context, SDN policies are decoupled from the physical layer and can be applied based on application-layer attributes.

In our work we have focused on detailing the main characteristics a cognitive module can have in the context of overcoming security failures in SDN, assuming it is implemented in the application plane, interacting with the controller through the northbound interface. While addressing the dynamic specificity of the autonomous SDN management, two major components emerged: the control loops (for which the literature acknowledges a panoply of answers) and the cognitive mechanism, that can be depicted through various particle like methods. A preliminary study has been conducted in understanding the ways of integrating cognition in an SDN context from the self-optimization perspective of the autonomous SDN management.

In order to illustrate the feasibility of cognition integration, a dynamic multi-objective context has been considered and a real use case has been identified as a possible scenario where the suggested approach can be applied in the dynamic context. Finally, the possible factors triggering dynamic changes in an SDN context have been identified and ways of including them in a multi-objective formulation also proposed.

The rest of the paper is organized as follows. Section 2 first provides an overview of the main security challenges raised up in SDN. Then, the non-cognitive functions/methods that may help in monitoring and detecting possible anomalies and attacks, are presented. Also, some security solutions built on these functions, already available in literature, are presented. Section 3 introduces the concept of cognition, and the use of cognitive mechanisms in conjunction with control loops in addressing the security concerns raised in the autonomous SDN management for a centralized SDN context. Section 4 details the use of multi-objective dynamic formulations for improving SDN security. To show the potentiality of the suggested approach, we illustrate how it can be applied in a real use case. Finally, Section 5 concludes the paper and presents possible paths for future research.

2 SDN Non-cognitive Functions for Enabling Security

Software Defined Networking with its new paradigm has introduced new edges and facets in the network, that may open a way to spiteful attackers [33]. In detail, the weaker points of the SDN architecture identified as being easily exploitable by attackers were: (i) the software applications-based control of the network, and (ii) the fully centralized control (i.e., single controller) of the network that represents *a single point of failure*.

Extensive analysis studies of

The newly introduced attack vectors where subject to extensive analysis studies, see [33]. Among the newly introduced attack vectors, it is worth to mention: Denial of

Service (DoS) attack on the controller from the SBI, attacks on SDN-enabled switches (in fact, the new protocols they support may imply new vulnerabilities), attacks using control channel (i.e., communication between controller and switches), attacks on the controller itself as well as attacks on the applications (with deployment of malicious applications) running on the controller.

Beyond the aforementioned security challenges that it has introduced, it has to be emphasized that SDN has also brought a plus to the whole system, i.e., the ability (of the SDN controller) to react in *real-time* to anomalies [41, 54] thanks to the programmability of the network. Several factors can trigger anomalies. In detail, based on the type of dynamic changes that take place in the network, and the location of the security breaches, anomalies can occur at various levels of the network. We can distinguish changes in the network configuration, in the traffic's behavior, and finally in the users' profile. It has to be noticed that behind every anomaly/change taking place in the network, an attack could be hidden.

SDN controllers provide functionalities which enable to cope with such, very often undesirable, anomalies. A set of useful features, able to enhance the overall security of the system [57], are already available in open-source controllers, like Floodlight [18], OpenDayLight [45], POX [49]. Hereafter we refer to them as *non-cognitive functions* because they provide basic functionalities for supporting security, without the use of cognition.

In the following sections we first introduce the non-cognitive functionalities, which from our point of view, are potential important building blocks for SDN security applications. Then, we survey some security solutions that have been already proposed in literature, and built on the aforementioned type of functionalities.

2.1 SDN Controller Functionalities

In this section we list a set of OpenFlow-based SDN features that can be exploited by the controller applications for dealing with failures and anomalies, and thus, for providing network security. It is worth to notice that these features are ready to use in most of the open-source SDN controllers.

Network Topology. One of the most outstanding assets of SDN is the global view of the network. The information about all network elements, the links established among them, and the changes occurring in the network, are constantly monitored. By doing so, the controller tries to keep an up-to-date imagine of the network graph.

For the controllers based on Beacon technology [15], like OpenDayLigth, Floodlight and Beacon itself, this function is performed by the Topology Manager bundle. Other components such as ARP handler, Host Tracker, Device Manager, and Switch Manager, provide information to complete the topology database for the Topology Manager. Also the POX controller includes a topology module dealing with network topology.

Switch Statistics. According to the OpenFlow standard [38], the switches can maintain counters for several entities, such as: flow table, flow entry, port, queue, group, etc. For instance, they may count the number of received and transmitted packets per port, number of packets lookups in a flow table, number of active entries of a flow table, etc. The counters provide useful statistics that can be used for profiling network elements. The profiles can further be used in anomaly detection techniques. In OpenDaylight and Floodlight, the statistics are available through a *StatisticManager* class.

Flow Table Manipulation. The OpenFlow standard [38] allows to manipulate packets on per-flow basis [47]. Thus, it is possible to add/remove VLAN tag of the packet, set the transport port, set source and destination data layer and network addresses, set the next hop, drop the packet, read flow information, etc. In the OpenDayLight API, an Action class allows to manipulate flows, and take actions on them, through methods of a Flow class.

The per-flow manipulation is an important SDN feature that can be exploited in several way, for taking security countermeasures. For instance, it allows to:

– *Monitor flows*
  A controller application may store and analyze the aforementioned flow fields. For example, it is feasible to keep track of all ports on which given host tries to access another, thus, giving a possibility to detect port scanning which is usually used in cyberattacks [59]. Another example is to aggregate the incoming flows to a given host: abnormally high number of connections from different sources may identify a DDoS attack. Alternatively, the flow monitoring may be used to build host traffic profile for further anomaly detection.

– *Mirror traffic*
  To the end of Deep Packets Inspection (DPI), the controller may dynamically enable mirroring of a suspected flow at any switch and forward it to a specialized host.

– *Operate firewall at a switch*
  Basic firewall functionalities, such as Network Address Translation (NAT) [58], denying/allowing traffic by destination port or by source address can be executed directly in the switch, by installing the corresponding rules into the device. If the network manager or an SDN application needs to block traffic coming from a particular source, it is interesting to notice the possibility of blocking an unwanted traffic (e.g., from a malicious host) at the source (by installing the blocking rule directly at the switch to which the malicious host is connected) in order to avoid unnecessary additional network load; this is thanks to the global view of the network.

Direct Packets Injection. To verify whether the behavior of some switches is different from the one expected, it might be useful to inject a packet to the network and track its path/content. OpenFlow already supports such functionality that allows the controller to send a packet out through the datapath. For instance, in OpenDayligth there is a transmitDataPacket function within the DataPacketService class.

2.2 SDN Security Solutions Built on the Controller Functionalities

In literature some security solutions based on the security-oriented functionalities, presented in the previous section, have been already proposed. Hereafter, we overview them, underlying which SDN features they employed the most.

One of the classical solution for enabling a secure network consists in the implementation of enhanced monitoring mechanisms, which besides providing measures on the status of the traffic flows, are able to detect anomalous network behavior. For instance, the Distributed DoS scheme (DDoS) proposed in [8], employs, on top of the switches monitoring system, Self Organizing Maps by using *flow statistics* and *flow manipulation* functions, to detect potential anomalous flows. Another DDoS detection system, namely Defense4All [51] has been built into the OpenDaylight controller. By using a combination of *flow statistics, flow manipulation*, and *network topology* functionalities, it generates a standard traffic profile that compared with the current one allows to detect suspicious traffic. The latter is sent to a middle-box for more detailed analysis.

In turn a cost optimized monitoring mechanism based on genetic algorithms has been proposed in [6]. This method, built on top of *network topology* and *flow manipulation* functions, exploits the fact that DPI engines can be virtualized and dynamically deployed as pieces of software on commodity hardware. Jin et al. [28], by taking advantage of *flow table manipulation* and especially *flow monitoring* features, have provided a mobile malware detection system. The solution that they have proposed is able both to identify suspicious network activities through real-time traffic analysis, as well as to impose new security rules in real time, when needed.

To the aim of securing a software defined network, solutions involving more advanced hardware components, as programmable switches have been also proposed, as the Resonance system [42], which applies dynamic access control. In Resonance, the programmable switches are used to enforce dynamic access control policies based on both flow-level information and real-time alerts. In other words, *flow manipulation*, *flow monitoring* and *firewall* features are being used in order to reinforce SDN security.

SDN controllers are enabled also with a set of functions that allows the redirection of affected flows (e.g., flows manipulated by attackers) and the application of changes in the rules of faulty network elements. Solutions making use of these functions can be based on *network topology, flow manipulation* and *flow mirroring*, as for example in [56]. The suggested SDN-based framework provides monitoring services for large and dynamic cloud networks. Moreover, it automatically changes the routing paths of network flows, by transmitting them through network nodes where security devices reside. This traffic redirection approach has been used also in OpenSAFE system [4] which enables the arbitrarily direction of the traffic to security monitoring applications at line rates, by using *flow manipulation* and *mirroring* functions.

In the FlowTags architecture [17], in order to keep flow traceability, security devices add to outgoing packets Tags that provide additional contextual information with a traffic flow (e.g., source hosts or internal cache/miss state). These Tags are then used by SDN switches and (other) security devices for systematic policy enforcement. A similar approach has been also applied in SIMPLE [50], which introduces policy enforcement layer based on SDN *network topology* and *flow manipulation* features to efficiently steer network traffic. The advantage of SIMPLE solution in comparison to the

FlowTags one is that no modification to legacy security devices and existing SDN interfaces is required.

Finally, active security approaches have also been proposed, as in [24]. The framework introduces programmatic control within a novel feedback loop to control the detection of attacks, data collection for attack attribution, configuration revision, and reaction to attacks. All these operation are performed by exploiting *flow manipulation* features. Hand et al. built an initial prototype that extends the FloodLight SDN controller [18] to automatically interface with the Snort [52] intrusion detection system in order to detect anomalies.

3 ANM and Control Loops: The Ancestors of Cognitive Algorithms for SDN

Due to the increased complexity of networks, the huge amount of devices and users to handle, and the resulting challenge to efficiently control/manage and operate such kind of networks, the need of *Autonomic Network Management* (ANM) has raised up more and more in the last decade [1,34]. Based on the concept of *autonomic computing*, introduced for the first time by IMB in 2003 [25], ANM allows to set-up self-managing networks that are able to detect, (re)act and reconfigure them self, and thus, recover from undesired network status (e.g., failures). The autonomous approach has paved the way to the Software Defined Networking paradigm, whose final aim is to simplify the network management, by introducing network programmability. In fact, the ANM Autonomic Manager can be seen as an SDN controller, or can be defined through the applications interacting with the controller(s) at the level of the Northbound interface.

The self-management implies adaptation to dynamical changes, that trigger new challenges, not covered by existing prevention rules, from where the need of enabling the system to learn and react based on reasoning. Cognition can be considered as the nesting view, as it encloses amongst the various facets the learning and reasoning aspects and thus answers this challenge, as outlined for ANM cases [13].

Many different models have been proposed in literature for enabling cognition in a general network management context, but *can they be applied to software defined networks, given the analogy between SDN and ANM*, mentioned before*?* This is one of the issue we have addressed in the present work. Although not named as an SDN, but rather as a *knowledge plane for the Internet*, the ideas behind enabling cognition inside an SDN context, were already conceptually outlined [13]. The proposed cognitive skeleton relies on a distributed and decentralized perspective, different from the centralized approach adopted in the SDN architecture[1].

Following the definition provided by Clark [13] and later Mitola [40], a cognition enabled SDN represents a *cognitive network* that implies the existence of a cognitive process, aware of the overall current network status, and thus, able to react based on the existing conditions. The place of cognition in a generic centralized SDN environment can be foreseen at various planes, the most natural one being the application one,

[1] It has to be noticed that the SDN architecture can be also decentralized by using several controllers within the control plane, but this is out of the scope of our work.

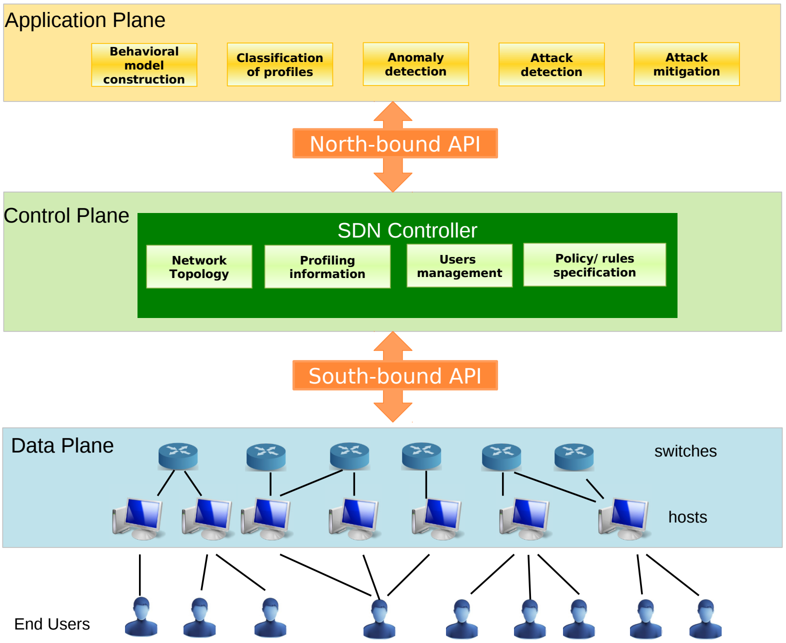


Fig. 1. A cognitive SDN architecture for security enforcement

as shown in Figure 1. The latter depicts the use of cognitive algorithms (e.g., machine learning techniques, or other probabilistic model building and profiling technique) in SDN security applications. It is worth to notice the clear separation made between the **control plane**, playing the role of knowledge keeper - gathering the information reflecting the status of the network, through the use of non-cognitive functionalities - and the **application plane**, intended to provide cognition based capabilities. As example of usage, when considering the Machine Learning (ML) as cognitive paradigm to be employed, the behavioral model construction translates in a machine learning model training.

3.1 Cognition as an Answer for Security Concerns in SDN

Following the time-line view [19] and the cognitive reasoning one (ignoring the time-line aspect) [32], we have built a self-optimization-based overview variant, depicted in Figure 2. Facing a security perspective, the cognitive reasoning, according to the level of granularity that is applied and the amount of information used, spans through three levels: *strategic*, *tactic* and *reactive*. In our case the classical time-line view approach [19] has been replaced by the reactive one, as this can be readily applied in the security context, through the controller non-cognitive functionalities (see Section 2) or for example, through the network components internal rules. It is worth noticing that by taking a traffic engineering perspective, the environment faced is intrinsically dynamic

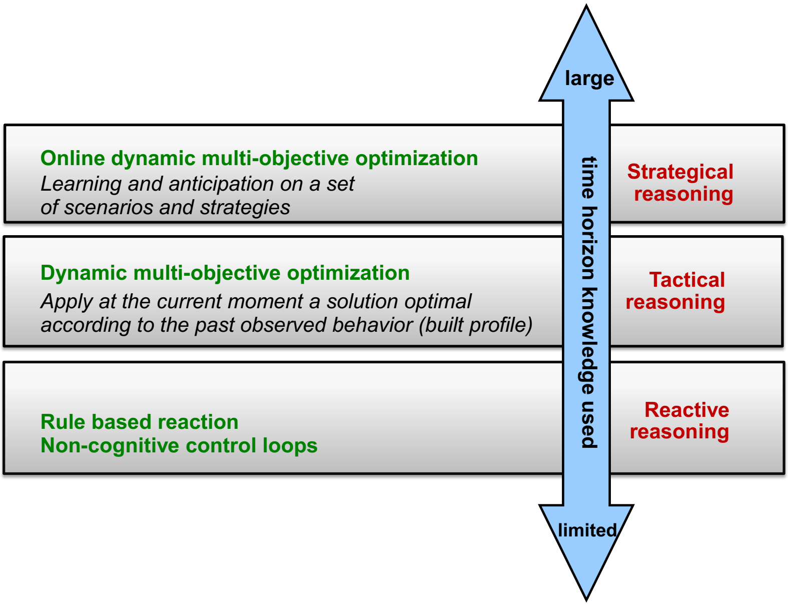

large
Online dynamic multi-objective optimization
Learning and anticipation on a set of scenarios and strategies
Strategical reasoning
Dynamic multi-objective optimization
Apply at the current moment a solution optimal according to the past observed behavior (built profile)
Tactical reasoning
Rule based reaction
Non-cognitive control loops
Reactive reasoning
time horizon knowledge used
limited

Fig. 2. Cognition approaches applicable for the SDN context from a security perspective, according to the information on the time horizon used

and as such the presented path. The left-hand side of Figure 2 outlines the possible models that apply in this context and the needed sub-components in order to provide the appropriate level of reasoning, depicted on the right-hand side.

Each of the covered layers (strategic, tactic and reactive) brings its own specificity and the level of complexity of the applied techniques increases accordingly:

Reactive reasoning: solely based on feedback control loops, implies the use of the non-cognitive security functionalities.

Profiling (classification based): useful in filtering data and data dimensionality reduction; constructs a probabilistic model from the observed behavior history of the monitored components (applies for the strategic and tactic reasoning).

Anticipation: applies only for the *strategic reasoning*. Implies the use of heuristic approaches or more advanced learning-based approaches applied in a dynamic context. Involves the use of several test scenarios and strategies.

An illustrative example on the use of cognitive reasoning for the prevention and mitigation of anomalies in an autonomous SDN management setting is provided in Section 4.

3.2 Main Components of Cognitive Approach

As in the autonomous network management, besides the representation of the problem's knowledge, the two main components that define cognition in an SDN context are: (1) **a cognition mechanism** (defined, e.g., through a particle-like method), and (2) **a control loop component**.

As a non-cognitive alternative, control loops can be developed as stand-alone solutions (i.e., without the use of any cognition), although with a limited range of applicability and covering deterministic (already observed) security threats. The confines of security issues they address - in the intent of providing the base functionalities enabling self-management - lay from self-configuration, to self-healing, self-protection or self-optimization. On the other hand, coming from control theory, control loops can provide also the proper environment for employing cognition in the SDN context.

Control loops are implemented by Autonomic Managers (AMs) that can operate at two different layers of the ANM architecture. In detail, AMs can work as *touchpoints* that directly manage the network components through monitoring and reacting/forwarding interfaces; or as *orchestration* mechanisms that coordinate the behavior of multiple AMs. As detailed in Table 1, several control loop mechanisms have been developed during the years, having different levels of applicability, and it is possible to distinguish between: *local control loops* that are applied in a specific part of the network or on a specific group of users; and *global control loops*, applied to the whole network and all its components.

Amongst the specific functionalities offered by the aforementioned environments to the security concerns, it is worth to notice, for example, that the 4D architecture encloses detection of forwarding loops and avoiding routing anomalies (persistent forwarding loops and permanent route oscillations). On the other side, MAPE [25] allows attack mitigation, through a self-protection functionality, enabling reaction to unauthorized access and use, virus infection and proliferation or denial-of-service attacks. In MAPE the control loops are classified according to their role in four categories: self-configuring, self-healing, self-optimizing and self-protecting.

FOCALE brings as a plus an orchestration effort, in which the cognition perspective has been integrated also through a fusion model that makes use of information models, ontologies, machine learning and reasoning.

Table 1 is not intended to exhaustively cover the domain, as various other projects exist, although with no specific focus on security aspects or intended for the autonomous management case. For instance, Haggle [22] and GENI [20] developed intrinsic control loops, intended mainly for the reliability and maintenance of the system, which can also be useful to some extent in the security context.

3.3 Cognitive Mechanisms

Biological systems are by nature self-managed, and allow the organism to dynamically adapt to environmental changes. Therefore, bio-inspired autonomic networks have been developed, by applying some biological principle to network management [36]. In the artificial intelligence area several techniques coming from Bayesian probability, evolutionary computation or machine learning areas can be employed.

Table 1. Control loops (CL) addressing security concerns and autonomous management challenges

Name, reference	Area of knowledge covered		Type of failures covered		Security aspect covered	CL specificity	Cognition
	local	global	single failure	multiple failures			
OODA[a], Boyd (1987) [7]	✓		✓		Attack prevention	*De facto* standard	-
4D architecture, (2005) [21]	✓		✓		Anomaly detection	Protocol level	Heuristics
MAPE[b], IBM (2006) [25]		✓	✓		Attack mitigation	Uses a shared knowledge base	Classification: association rule mining
FOCALE[c], Strassner *et al.* (2006) [60]	✓	✓	✓		Attack, intrusion prevention (based on OODA)	Policy driven, Network level, cognitive	Finite-state-machine
ConMAN, (2007) [3]	✓		✓		Intrinsic functionalities	Protocol abstraction enabled	Classification
GANA, (2013) [16]		✓	✓	✓	Data poisoning prevention, anomaly detection and prevention	Hierarchical	Three levels of cognition apply: strategic, tactical and classical
CogMan, Sungsu *et al.*(2013) [32]		✓	✓	✓	Failure recovery	Priority based CL	Fast Flow Setup algorithm

[a] Observe-Orient-Decide-Act (OODA).

[b] Monitor-Analyze-Plan-Execute (MAPE).

[c] Foundation-Observe-Compare-Act-Learn-rEason (FOCALE).

Machine learning as applied in the more general context of security in networking, can take several facets from *diagnosis* **(detection)**, to *maintenance* **(reaction)** or *prognosis* **(anticipation)**. As for the the detection task this can be focused on anomalies [37], attacks [11] or even intrusion detection [12], while considering cases where no information or scarce information is available. In the management of an autonomous network unsupervised learning represents the solution to be applied in a dynamic context, while supervised learning can be seen as an *a priori* task for detecting patterns that are to be implemented in the control loop.

Various highly modulable platforms are available in the community in order to support the development of such cognitive techniques. Only in the evolutionary area, one can notice various flavors based on a variety of languages as C/C++ for ParadisEO [9] or JAVA for ECJ [63], with an additionaly designed Multi-agent simulator of Networks (MASON) [35]. These are useful assets in coping with the optimization side of the problem. Also the machine learning side is well represented through data mining platforms, such as Weka [23].

4 Self-optimization in a Dynamically Changing SDN Environment

Static optimization theory has already been considered as a potential solution in inhansing the performances and management of classical networks. Nevertheless one of the main drawbacks, as compared with regret minimization, was related to the unhandled online aspect. Through the current proposal we aim at providing a possible modeling that overcomes this aspect, by simultaneously handling multiple objectives in an online dynamic manner, inside an optimization context. The multi-objective area enables inclouding a set of conflicting or different by nature objective functions to be handlded simultaneously (e.g. one following the global network-wide perspective and one maximizing the individual gain), thus answering the local to global pefromance perspective raised in [2].

As outlined in the previous section, self-optimization represents one of the main directions enabling improvements in the security of a self-managing SDN. In this section we present a preliminary study on how the so-called **strategic reasoning** (as depicted in Figure 2), and in detail, dynamic multi-objective optimization can be applied in an SDN context. Finally, the feasibility of using multi-objective formulation is also investigated in a specific SDN use case, taken under analysis.

4.1 Multi-objective Dynamic Formulation

The first step to design dynamic self-optimization consists in identifying the factors that trigger complexity. In the SDN context, there are several layers where dynamic changes can occur, triggered by

(a) External factors
- Environmental changes affecting the network functioning, such as intrusion or attacks.

(b) Internal factors
- **Network configuration changes**, leading to *anomalous behavior*, involving:

  - the physical network configuration;
  - the switches rules;
  - the applications connected to the controller;
  - the controller behavior (e.g., priority of applications).
- **Traffic changes**, leading for instance to *congestion*.
- **Users changes**: addition/removal of users, mobility of users, remodeling of user groups.

In order to model the dynamic behavior based on the outlined dynamic factors we introduce a dynamic multi-objective problem formulation. Let t be a monotonically increasing value on a time period $[t_0, t_{end}] \in \mathbb{R}^+$ and let σ be an environment derived set of parameters, uncontrollable and external to the system (i.e. the students' anual courses timetable influencing their network usage - the volume and type of requests submited according to the various periods: exams, holidays, etc.). The behavior of a dynamic multi-objective optimization problem (system) for the SDN context can be depicted by $H(F_\sigma, D, x, t)$, where

F_σ multi-objective support function;
D time-dependent functionals modeling the dynamic behavior of a specific component of the system;
x the set of variables / parameters.

The objective functions, defined as

$$F_\sigma : X \to Y, F_\sigma(x) = [f_{\sigma,1}(x), \ \dots \ , f_{\sigma,k}(x)] \tag{1}$$

evolve in time according to

$$\int_{t_0}^{t_{end}} F dt = \left(\int_{t_0}^{t_{end}} f_1 dt, \ \dots \ , \int_{t_0}^{t_{end}} f_k dt \right) \tag{2}$$

The most complex setting to be handled in this context comes from the presence of external factors (e.g., attackers) triggering complexity. In this case we are facing an online dynamic multi-objective formulation - a 4^{th} order formulation, following the classification proposed in [61] - that can be depicted as

$$H(F_\sigma, D, x, t) = F_{D(\sigma,t)}(x, t), \tag{3}$$

leading to a formulation of the type

$$\arg\min_{\mathbf{x}(t)} \int_{t_0}^{t_{end}} F_\sigma(\mathbf{x}(t), t) \text{ dt} \overset{def}{=} \min_{\mathbf{x}(t)} \left\{ \left(\int_{t_0}^{t_{end}} f_{\sigma,i}(\mathbf{x}(t), t) \text{ dt} \right)_{1 \le i \le k} \right\}$$

In this case the external parameters are modeled through the $D(\sigma, t)$ and can be represented through context dependent probability distributions, for which the parameters need to be estimated. In order to provide competing results a set of scenarios and strategies needs to be defined. The scenarios can be based on existing attack patterns, e.g., Address Resolution Protocol (ARP) poisoning.

When tackling scenarios from the traffic engineering perspective, the network can be modeled as a graph, with x enclosing the representation of the network topology (e.g., through adjacency matrix, etc.). Next, scenarios can be built by drawing samples from, for example, a power law distribution with random graphs modeling different instance types, as the case in [39]. In this mentioned work, besides the random graph (topology) sampling, each edge is modeled by a Bernoulli random variable. In order to deal with sparse graphs, the authors rely on a log-linear model for edge probabilities which, w.r.t. specific context aspects, can be seen as a generalized linear model.

Regarding the objective functions to be used, they can be related to the quality of the flow clustering, required in determining traffic patterns, useful for both internally and externally driven dynamic problems. One possible scenario making use of the flow classification for reinforcing the security in SDN can cover the anomalous switches behavior, as described in the following section.

4.2 Secure SDN Use Case: Mitigation of Anomalies Due to Misbehaving Switches

In a centralized single-controller SDN architecture, the controller has the role of mitigating anomalies by means of specific applications, connected with the controller through the Northbound interface (Figure 1). We assume one of the applications consists in a dynamic multi-objective optimization technique, using the problem definition for anomaly mitigation as depicted herein.

The investigated security concern is induced by anomalous behavior of switches. Although, there exist frameworks such as, e.g., FRESCO [57] that help in preventing conflicting rules to be applied at a switch level, in an SDN network, switches can be under the influence of malicious users/attackers for different reasons. First of all, the use of the Transport Layer Security (TLS) protocol [14], for protecting the channel between the controller and the switches, is optional according to the OpenFlow specification [46]. Beside that, some attack patterns are known *a priori*, such as ARP poisoning, DHCP snooping, broadcast/multi-cast rate limiting, MAC address limits. This allows to identify which network properties should be monitored (e.g., the number of flows per switch).

Switches in an SDN network can deviate from the controller tasks in a number of ways including: redirecting or delaying traffic, dropping or modifying packets, falsifying statistics or requests sent to the controller. According to [5], the only way to address this problem is to dump and inspect the flow tables on a per-switch basis which is a very resource consuming procedure. An alternative consists in learning from the past behavior of the switches in terms of traffic flow transiting the switch. This can be done through *profiling techniques* and *unsupervised learning techniques*. Once the anomalous switch has been identified, the controller needs to mitigate the anomaly, by applying countermeasures at the traffic level.

The multi-objective formulation allows among others to apply solutions that address several security challenges simultaneously, that are either conflicting or different by nature and leaves place also for the possible integration of other criteria, such as performance related objectives.

In order to mitigate security concerns various probabilistic techniques have been successfully applied on simplified model formulation, or addressing clustering issues, see for instance, the Ant Colony Optimization approach proposed in [10] or the Symbolic Dynamic Filtering applied in [29]. These techniques can be seen as potential candidates to be exploited in our future work for the experimental evaluation of the proposed use case.

5 Conclusions

In this paper we have described non-cognitive SDN features which are potential important building blocks for SDN security applications. We have also showed how such non-cognitive functions provide support against different security attacks. As a follow up, we detailed on the intersection between control loops, autnomous network management and cognitive networking. Finally, we have suggested the use of cognition for enhancing the security level that can be achieved with the non-cognitive function, through dynamic multi-objective optimization. In our future work we aim at applying the outlined cognitive module to real use cases. Moreover, we will investigate how the multi-objective formulations can be used for providing not only security, but also network reliability and better QoS performance.

Acknowledgment. This publication is based in parts on work performed in the framework of the CoSDN project, INTER/POLLUX/12/4434480, funded by the Fonds National de la Recherche, Luxembourg. The authors would like to thank Gabriela Gheorghe for useful discussions on SDN usecase.

References

[1] Agoulmine, N., Balasubramaniam, S., Botvitch, D., Strassner, J., Lehtihet, E., Donnelly, W.: Challenges for autonomic network management. In: Proc. of 1st IEEE International Workshop on Modeling Autonomic Communications Environments, MACE 2006 (2006)

[2] Avramopoulos, I.C., Rexford, J., Schapire, R.E.: From optimization to regret minimization and back again. In: Fox, A., Basu, S. (eds.) SysML. USENIX Association (2008)

[3] Ballani, H., Francis, P.: CONMan: A step towards network manageability. SIGCOMM Comput. Commun. Rev. 37(4), 205–216 (2007)

[4] Ballard, J.R., Rae, I., Akella, A.: Extensible and scalable network monitoring using OpenSAFE. In: Proc. of the 2010 Internet Network Management Conf. on Research on Enterprise Networking, INM/WREN 2010, p. 8. USENIX Association, Berkeley (2010)

[5] Benton, K., Camp, L.J., Small, C.: Openflow vulnerability assessment. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN 2013, pp. 151–152. ACM, New York (2013)

[6] Bouet, M., Leguay, J., Conan, V.: Cost-based placement of virtualized Deep Packet Inspection functions in SDN. In: Military Comm. Conf., IEEE MILCOM 2013, pp. 992–997 (November 2013)

[7] Boyd, J.: Destruction and Creation. Operational level of war, U.S. Army Command and General Staff College. Center for Army Tactics (1987)

[8] Braga, R., Mota, E., Passito, A.: Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: 2010 IEEE 35th Conference on Local Computer Networks (LCN), pp. 408–415 (October 2010)

[9] Cahon, S., Melab, N., Talbi, E.G.: Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. Journal of Heuristics 10(3), 357–380 (2004)

[10] Carvalho, L., Rodrigues, J., Barbon, S., Lemes Proenca, M.: Using ant colony optimization metaheuristic and dynamic time warping for anomaly detection. In: 2013 21st International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–5 (September 2013)

[11] Casas, P., Mazel, J., Owezarski, P.: Steps towards autonomous network security: Unsupervised detection of network attacks. In: NTMS, pp. 1–5. IEEE (2011)

[12] Casas, P., Mazel, J., Owezarski, P.: Unsupervised network intrusion detection systems: Detecting the unknown without knowledge. Comp. Comm. 35(7), 772–783 (2012)

[13] Clark, D.D., Partridge, C., Ramming, J.C., Wroclawski, J.: A knowledge plane for the Internet. In: Feldmann, A., Zitterbart, M., Crowcroft, J., Wetherall, D. (eds.) SIGCOMM, pp. 3–10. ACM (2003)

[14] Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard) (August 2008)

[15] Erickson, D.: The beacon OpenFlow controller. In: Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN 2013, pp. 13–18. ACM, New York (2013)

[16] ETSI: Generic autonomic network architecture (an architectural reference model for autonomic networking, cognitive networking and self-management). ETSI GS AFI 002 V1.1.1 (April 2013)

[17] Fayazbakhsh, S.K., Sekar, V., Yu, M., Mogul, J.C.: FlowTags: Enforcing network-wide policies in the presence of dynamic middlebox actions. In: Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN 2013, pp. 19–24. ACM, New York (2013)

[18] Floodlight project, `http://www.projectfloodlight.org`

[19] Gat, E.: On three-layer architectures. In: Kortenkamp, D., Bonnasso, P.R., Murphy, R. (eds.) Artificial Intelligence and Mobile Robots, pp. 195–210 (1998)

[20] Geni design principles. Computer 39(9), 102–105 (2006)

[21] Greenberg, A., Hjalmtysson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., Zhang, H.: A clean slate 4D approach to network control and management. SIGCOMM Comput. Commun. Rev. 35(5), 41–54 (2005)

[22] IST FP6 Haggle project, `http://www.haggleproject.org`

[23] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)

[24] Hand, R., Ton, M., Keller, E.: Active security. In: Proc. of the 12th ACM Workshop on Hot Topics in Software Defined Networking, HotNets-XII, pp. 17:1–17:7. ACM, New York (2013)

[25] IBM: An architectural blueprint for autonomic computing. ONF White paper (2006)

[26] IEEE SDN/NFV subcommittee, `http://cms.comsoc.org/eprise/main/SiteGen/TC_SDN_NFV/Content/Home.html`

[27] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., Vahdat, A.: B4: Experience with a globally-deployed software defined WAN. SIGCOMM Comput. Commun. Rev. 43(4), 3–14 (2013)

[28] Jin, R., Wang, B.: Malware detection for mobile devices using software-defined networking. In: Proc. of the 2013 2nd GENI Research and Educational Experiment Workshop, GREE 2013, pp. 81–88. IEEE Computer Society, Washington, DC (2013)

[29] Jin, X., Guo, Y., Sarkar, S., Ray, A., Edwards, R.: Anomaly detection in nuclear power plants via symbolic dynamic filtering. IEEE Transactions on Nuclear Science 58(1), 277–288 (2011)

[30] Katz, D., Ward, D.: Bidirectional Forwarding Detection. IETF Internet-Draft, draft-ietf-bfd-base-11.txt (January 2010)

[31] Khurshid, A., Zhou, W., Caesar, M., Godfrey, P.: VeriFlow: Verifying network-wide invariants in real time. ACM SIGCOMM Comp. Commun. Rev. 42(4), 467–472 (2012)

[32] Kim, S., Kang, J.M., Seo, S.S., Hong, J.W.K.: A cognitive model-based approach for autonomic fault management in OpenFlow networks. Int. Journal of Network Management 23(6), 383–401 (2013)

[33] Kreutz, D., Ramos, F.M., Verissimo, P.: Towards secure and dependable software-defined networks. In: Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN 2013, pp. 55–60. ACM, New York (2013)

[34] Louca, A., Mauthe, A., Hutchison, D.: Autonomic network management for Next Generation Networks. In: Proc. of 11th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, PGNet 2010, pp. 1–6 (2010)

[35] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: A multiagent simulation environment. Simulation 81(7), 517–527 (2005)

[36] Manzalini, A., Minerva, R., Moiso, C.: Bio-inspired autonomic structures: a middleware for telecommunications ecosystems. In: Vasilakos, A.V., Parashar, M., Karnouskos, S., Pedrycz, W. (eds.) Autonomic Communication, pp. 3–30. Springer US (2009)

[37] Mazel, J., Casas, P., Labit, Y., Owezarski, P.: Sub-space clustering, inter-clustering results association & anomaly correlation for unsupervised network anomaly detection. In: CNSM, pp. 1–8. IEEE (2011)

[38] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. Proc. of the ACM SIGCOMM 2008 Conference. 38(2), 69–74 (2008)

[39] Miller, B., Arcolano, N., Bliss, N.: Efficient anomaly detection in dynamic, attributed graphs: Emerging phenomena and big data. In: 2013 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 179–184 (June 2013)

[40] Mitola, J.: Cognitive Radio Architecture. In: Cognitive Networks - Towards Self-Aware Networks. Wiley London, London (2007)

[41] Nadeau, T.D., Gray, K.: SDN: Software Defined Networks, 1st edn. O'Reilly Media (September 2013)

[42] Nayak, A.K., Reimers, A., Feamster, N., Clark, R.: Resonance: Dynamic access control for enterprise networks. In: Proc. of the 1st ACM Workshop on Research on Enterprise Networking, WREN 2009, pp. 11–18. ACM, New York (2009)

[43] Open Networking Foundation: Migration use cases and methods. ONF White paper (December 2013)

[44] Open Networking Foundation: SDN architecture overview. ONF White paper (December 2013)

[45] OpenDaylight project, http://www.opendaylight.org

[46] OpenFlow Switch Specification (version 1.1.0), https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.1.0.pdf (February 2011)

[47] OpenFlow Switch Specification (version 1.4.0), https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf (October 2013)

[48] Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M., Gu, G.: A security enforcement kernel for OpenFlow networks. In: Proc. of the 1st Workshop on Hot Topics in Software Defined Networking, pp. 121–126. ACM (2012)

[49] POX controller, http://www.noxrepo.org/pox/about-pox

[50] Qazi, Z.A., Tu, C.C., Chiang, L., Miao, R., Sekar, V., Yu, M.: SIMPLE-fying middlebox policy enforcement using SDN. SIGCOMM Comput. Comm. Rev. 43(4), 27–38 (2013)

[51] Radware: Defense4All, User Guide, https://wiki.opendaylight.org/view/Defense4All:User_Guide (2014)

[52] Roesch, M.: Snort - lightweight intrusion detection for networks. In: Proc. of the 13th USENIX Conference on System Administration, LISA 1999, pp. 229–238. USENIX Association, Berkeley (1999)

[53] Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Core, RFC 6120. The Internet Engineering Task Force (March 2011)

[54] Scott-Hayward, S., O'Callaghan, G., Sezer, S.: SDN security: A survey. In: Proc. of IEEE SDN Conf. for Future Networks and Services (SDN4FNS), pp. 1–7 (November 2013)

[55] Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., Rao, N.: Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Comm. Magazine 51(7), 36–43 (2013)

[56] Shin, S., Gu, G.: CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?). In: Proc. of the 2012 20th IEEE Int. Conf. on Network Protocols, ICNP 2012, pp. 1–6. IEEE Computer Society, Washington, DC (2012)

[57] Shin, S., Porras, P.A., Yegneswaran, V., Fong, M.W., Gu, G., Tyson, M.: FRESCO: Modular composable security services for software-defined networks. In: 20th Annual Network and Distributed System Security Symposium (NDSS 2013). The Internet Society (February 2013)

[58] Srisuresh, P., Holdrege, M.: IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663 (Informational) (August 1999), http://www.ietf.org/rfc/rfc2663.txt

[59] Staniford, S., Hoagland, J.A., McAlerney, J.M.: Practical automated detection of stealthy portscans. J. Comput. Secur. 10(1-2), 105–136 (2002)

[60] Strassner, J., Agoulmine, N., Lehtihet, E.: FOCALE: A Novel Autonomic Networking Architecture. Int. Trans. on Systems Science and Applications 3(1), 64–79 (2007)

[61] Tantar, E., Tantar, A.A., Bouvry, P.: On dynamic multi-objective optimization, classification and performance measures. In: IEEE Congress on Evolutionary Computation, pp. 2759–2766. IEEE (2011)

[62] Vasseur, J.P., Le Roux, J.L.: Path Computation Ement Communication Protocol, RFC 5440. The Internet Engineering Task Force (March 2009)

[63] White, D.: Software review: the ecj toolkit. Genetic Programming and Evolvable Machines 13(1), 65–67 (2012)

[64] Xie, H., Tsou, T., Lopez, D.R., Yin, H.: Use Cases for ALTO with Software Defined Networks. IETF Internet-Draft, draft-xie-alto-sdn-extension-use-cases-01 (January 2013)

[65] Yang, L., Dantu, R., Anderson, T.A., Gopal, R.: Forwarding and Control Element Separation (ForCES) Framework, RFC 3746. The Internet Engineering Task Force (April 2004)

Application of Cognitive Techniques to Network Management and Control

Sławomir Kukliński[1,2], Jacek Wytrębowicz[1], Khoa Truong Dinh[1], and Emilia Tantar[3]

[1] Warsaw University of Technology, Warsaw, Poland
slawomir.kuklinski@tele.pw.edu.pl, jwt@ii.pw.edu.pl,
k.truongdinh@stud.elka.pw.edu.pl
[2] Orange Polska, Poland
[3] University of Luxembourg, Luxembourg
emilia.tantar@uni.lu

Abstract. This paper describes the latest communications technologies emphasizing the need of dynamic network control and real-time management operations. It is advocated that many such operations can profit from cognitive learning based techniques that could drive many management or control operations. In that context a short overview of selected networking approaches like 3GPP Self Organizing Networks, Autonomic Network Management and Software-Defined Networking, with some references to existing cognitive approaches is given.

Keywords: machine learning, SON, LTE, SDN, autonomic network management, artificial intelligence.

1 Introduction

In recent years a dynamic growth of communication networks can be observed. This growth concerns especially commercial IP networks and mobile systems, like UMTS (Universal Mobile Telecommunications System) or LTE (Long Term Evolution). Not only the number of network nodes is growing, but also network customer base as well as number of offered services. The services have differentiated traffic requirements and many of them, like video services and cloud processing, require high-speed links and high reliability. The users mobility is another problem that has to be taken into account - the traffic demands change in time and move between different geographic areas. The new situation requires a change of the paradigm of network deployment and maintenance. In the past the deployment of new networks was preceded by careful design and planning in order to optimize long-term investment and at the same time to satisfy present and potentially future (predicted) end-users needs. Unfortunately, the above-mentioned growth of users, services, and traffic related demands make the planning based approach no more efficient or applicable. The solution to this problem is real-time adaptive control of networks, which should enable dynamic allocation of resources and global optimisation of network usage. In response to such demand, concepts like SDN [32], i2rs [5] or GMPLS [31] have arisen. These concepts, implemented already in some networks, are still under evolution; but what is important, they make network control and management programmable. The difference between network control and

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_7, © Springer International Publishing Switzerland 2014

management lies in their purpose: the control is responsible for fast decisions related typically to users behaviour, network management is responsible for network operations and is much slower than network control.

There is however another problem related to the dynamic growth of network nodes, services and end-users. This problem is related to the old-fashioned way in which networks are operated and managed at present. Operators manage networks to configure them properly, handle faults, optimize performance and to cope with security issues. Actually such management is quasi-static, centralized with a key role of human operator. So far only some management operations are automated. The growth of network nodes, services and end-users makes such centralized, human centric management slow, error prone, inefficient and expensive. To cope with this problem, the concept of so-called Autonomic Network Management (ANM) [3] has been born. The concept lies on continuous measurements of the network and execution of suitable decisions according to the network state. Such feedback loop based management has been a subject of investigation of many European projects [9,13,42,47] and is now under standardization in ETSI [18]. The issue of self-management (a synonymous of autonomic network management) is also a hot topic in 3GPP (3rd Generation Partnership Project). In latest generation of mobile systems the concept Self-Organizing Networks (SON) has been proposed [46]. The goal of SON is to automate some of Radio Access Network (RAN) related management operations. This autonomic network management requires fast and adaptive to network state decisions.

The above mentioned technologies like SDN, i2rs, ANM, SON require algorithms, which realize predefined goals of network management or control. It means that a control theory or other real-time optimization techniques are required for their implementation. Fortunately the already mentioned approaches provide the programmability, which is possible not only by the solution manufacturer, but also by network operators, 3rd parties or even by end-users. It should be noted however that the above-mentioned operations are complex ones to realize control and management goals, many (if not all) network devices have to be involved and the complexity of such devices is typically high. Another complication is introduced by the independent optimization of multiple goals. Due to conflicting goals such approach in many cases leads to conflicting control or management decisions. For example minimization of energy consumption is typically against maximization of network performance. Simultaneous optimization of all goals is not being realized by todays management solutions.

There is no doubt that the control theory based or heuristic algorithms can be applied to network control and autonomic management. However, due to complex network behavior, the selection of algorithms or tuning of their parameters is in many cases hard or impossible. A solution of the problem is to use cognitive approaches, i.e., algorithms that have learning abilities. Such approach is very attractive, but also comes with the problem of learning convergence, and the problem of trust to the algorithm behavior. It has to be noted that networks have to operate reliably. Therefore any uncertainty related to algorithm behavior has to be minimized. Despite such constraints, the learning algorithms should be evaluated in the context of network control and management, and such research activity is already observed [9,13,15]. In this paper we want to provide an overview of cognitive techniques, which potentially can be applicable to solve network control and management problems. A short overview of cognitive techniques with their specific properties is presented in Section 2. Section 3 describes areas of applications of cognitive techniques, distinguishing radio Self-Organized Networks, autonomous fixed networks, and Software-Defined Networking. The section gives also some proposals of using cognitive techniques, and if applicable, examples of applications where such techniques are already in use. Finally Section 4 concludes the paper.

2 Overview of Cognitive Techniques

The goal of the section is to list some well-known cognitive techniques, which could be applicable to solve networking problems. Each algorithm will be described shortly from the implementers point of view, emphasizing key features of its application.

2.1 Selected Cognitive Techniques Outline

Cognitive techniques include a big set of algorithms, which common denominator is learning. They are so-called neural networks, evolutionary algorithms, machine learning, knowledge based reasoning and many other techniques. The common feature of them is an examination of the environment and taking subsequent actions by analyzing results of previous decisions. Algorithms may learn in different ways. In some of them the learning phase precedes the normal operations in such case we talk about off-line learning. In some cases we have on-line learning. The latter case can be split into continuous learning, in which the algorithm can re-learn; and the case, when during initial phase of operations the algorithm can both operate and learn, and has no re-learning abilities later on. For the off-line learning a set of training data has to be prepared. Such set is typically extracted from real data.

In so-called supervised learning, the training set consists of correct outputs to all input vectors included in it. The error between expected and actual output drives the learning procedure. The learning is iterated until for the whole training set the output error will be reduced to a satisfactory value. An example of a technique, which uses supervised learning, is multi-layer perceptron [39]. A multi-layer perceptron is a feedforward artificial neural network, based on multiple layers, of so-called neurons. Neurons are elements that compute a scalar product of their inputs and their respective weights; finally the scalar product is the input of nonlinear function (i.e., neuron activation function), which typically is sigmoid. The learning is performed using the back propagation (gradient descent) technique. The goal of the learning is to define neurons weights. Learning starts typically with randomized weights and is known to be slow. Normal operations of multi-layer perceptrons are generally fast, i.e. decisions are taken in one pass only. It is expected that after learning the perceptron will properly respond to unknown input, i.e. to the input that is not included in the training set. For proper behaviour of the network it is important to provide a representative set of input-output pairs in the training set. There are no strict rules for the design of multi-layer perceptron, especially in regard to the number of so-called hidden layers and their size. The key parameter of the learning, so-called learning rate may impact learning convergence and results. The re-learning is not possible in this method. Typically multi-layer perceptron is used as a classifier.

Another technique that uses supervised learning is called Support Vector Machines [14] used for linear or non-linear classification and regression analysis.

In opposite to supervised learning also unsupervised learning exists. In such approach the training set should contain only input data, no input-output pairs as in supervised learning are used. This type of learning is used to find the hidden structure of the input data set, and is closely related to statistical density estimation. The best-known approach of such learning is Self-Organizing Maps concept [28]. This technique is fast, but the analysed data set cannot be dynamically changed, it has to be prepared in advance. In case of changes in the input set, the algorithm has to restart. Unsupervised learning is typically used for data clustering. In recurrent neural networks, like Hopfield networks [23], the output of the neuron, which is similar to that used in multilayer perceptron, but typically has a binary activation function, is fed back to its inputs with predefined weights as well as the input vector. During iterations (synchronous or asynchronous ones) this type of network goes towards equilibrium (if certain conditions are satisfied). The Hopfield network, which weights are predefined by the Hebb rule, is typically used as Content Addressable Memory (CAM). CAM is able to restore incomplete input patterns. For the Hopfield network an energy scalar can be assigned, and during iterations the energy cant increase, and typically should go down. This property is used for application of the Hopfield network for solving optimization problems, like the travelling salesman problem. The main problem of the application of this network is the convergence time and potential lack of stability.

Reinforcement Learning (RL) is a type of machine learning technique that exploits the learning agents experience in order to learn the optimal behaviour in an environment [44]. Through its interaction with the environment, the agent tries to learn actions for particular states of the system, so that the long-term rewards are maximized. RL learns its actions on basis of the observation of the environment; however each action impacts the environment. RL performs an online search to find an optimal decision policy in multi-stage decision problems. The learning problems are usually modelled as Markov Decision Processes [40] and solved using Dynamic Programing techniques. In opposite to other learning approaches, RL evaluates the taken actions rather than corrections to actions. In combination with Fuzzy Sets Theory [49], the RL-FIS (Fuzzy Inference System) is widely applied in real world to different control processes, especially its model-free based learning variant (Q-learning). RL approach is typically used for continuous real-time learning. Multi-agent RL approaches can be used to simultaneously solve multiple problems.

In Case-based Reasoning (CBR) [2] possible actions are based on the past experiences in similar circumstances. For each case (input) the status of the corresponding environment as well as taken actions are memorized. During the analysis of a new case the prior cases are looked for similarity. If such similarity is found a corresponding, already memorized, action is taken. If there is no good match, a new action is created for such a new case. The typical CBR process lies on looking for the similarity with other cases and if found, taking appropriate action(s). If no similarity is found, that case is adapted and retained. For proper operation (correct generalization) the prior cases should be statistically relevant.

Particle algorithms [16] are an extensively used computational tool not only in engineering, but also in machine learning, statistics and physics. Nevertheless, various denominations have been developed simultaneously (according to the context where they were employed), ranging from: particle filters, bootstrap or genetic filters, population Monte Carlo methods, sequential Monte Carlo models, genetic search models, branching and multi-level splitting or particle rare event simulations. Evolutionary Algorithms (EA) [7] represent a heuristic approach inspired by natural evolution, having as main applicative area optimization. They include Genetic Programming [29,48], Genetic Algorithms [19] and Evolutionary Strategy [8]. According to [38], in the EA area the algorithmic development can be divided into two main paths: direct and indirect search methods. Direct methods rely on direct sampling of the objective function and the search is conducted by applying a set of perturbations on the initially generated set of samples (often randomly generated). Examples in this direction include: Hill-Climbing, Nelder-Mead, Solis and Wets, Tabu Search, Variable Neighborhood Search, Simulated Annealing, Iterated Local Search. Indirect methods suppose, in a continuous optimization context, the use of second order derivatives.

Besides the alrady mentioned single solution based approaches, set oriented methods emerged, their foundation being due to Holland [22]. Several variants exist, including among others, Scatter Search, Differential Evolution, Artificial Immune Systems or Swarm Intelligence, e.g. Bee Colony and Ant Colony Optimization. In addition to stand-alone algorithms, various hybrid versions were developed thus making place for meta-heuristics and later, hyper-heuristics. Another trend supposes the use of additional tools or information. Examples are the Covariance Matrix Adaptation [20], relying on information obtained from the landscape of feasible solutions; or Estimation of Distribution Algorithms [30] that construct a probabilistic model in order to capture the traits of solutions that fit a set of given criteria, e.g. fitness below a specified threshold. Multiple criteria can also be treated simultaneously, this being covered by the multi-objective optimization area. An extensive bibliography in the area is available at [12].

These techniques are applied mainly offline, nevertheless a specific branch of dynamic optimization handles contexts involving the use of dynamically evolving environments.

There are also other algorithmic techniques, which can be nicely combined with cognitive ones. Such techniques include principal component analysis [25], Fuzzy Sets [49], Rough sets [36], clustering techniques like k-means clustering [21], k-nearest neighbour clustering [26], Bayesian networks [24], Hidden Markov Models [37], Markov Decision Processes, or discovery of frequent patterns [10]. In general, using some pre-processing of input data may help in having a more efficient implementation of cognitive algorithms, reducing their learning time and improving accuracy.

2.2 Drawbacks of Cognitive Approaches

The main value of cognitive approaches is that network operators, who dont know how to solve a given problem or the problem is too complex to be solved using analytical tools, can use these approaches to find a solution. Although it should be noted that optimization and learning techniques will make use of a simplified model of the treated problem, thus inherently loosing some accuracy through the description employed.

However such approaches come also with some problems related to their proper application, and selection of many learning algorithm parameters. This includes for example so-called learning rate, exploitation/exploration parameter, neural network topology or initial pre-processing of input data. It has to be noted, that an expert knowledge can be used for input data pre-processing as well as for evaluation of results.

Another problem is related to the algorithm convergence time, especially when the algorithm is applied in real-time, although techniques with proven performance guarantees in theory and practice emerge, see [16]. In some recursive approaches there is potential danger of unstable (chaotic) behaviour of the algorithm. However the most important problem with many cognitive algorithms lies on the lack of trust that they behave properly if new situation will happen, and that they provide high overall quality of results in presence of uncertainty.

3 Areas of Applications of Cognitive Techniques in Networks

The cognitive techniques have huge potential; however until now their usage for solving communication networks problems (control and management) is rather low. In this section we outline potential application areas of cognitive techniques in communication networks. We also propose some candidate techniques for solving selected problems. If a cognitive approach has been already proposed, by a research paper or an implementation, this is also mentioned in this section.

3.1 Application of Cognitive Techniques for SON Enabled Mobile Networks

At present very dynamic development of mobile networks is observed. It can be also noted that a new generation of mobile systems is typically introduced in several years after the deployment of the previous one. The users of these networks require more and more bandwidth.

The most problematic part of each mobile system is so called Radio Access Network (RAN). A widely accepted approach to RAN lies on cellular structure of the radio network, and one hop communication between user terminals and one of the radio base stations. In Europe, average operators RAN is composed of thousands of radio base stations to provide nationwide coverage and appropriate capacity to handle users traffic demands. It is worth to note that in the latest generations of mobile systems, the typical cell size is smaller than in previous ones in order to handle more traffic; therefore the number of cell operated by the operators grows. The cellular structure of RAN imposes also the issue of users mobility handling, i.e. seamless switching users radio links between neighbouring cells without interruption of active communication sessions. The mechanism is known as handover.

At present every operator deploying new network has to configure properly each base station, and during its operation has to optimize RAN performance and handle possible faults. The configuration process is so far manual and static, performance optimization is very limited, and fault handling has limited automation level. It is worth to emphasize that manual management of such network is costly, error prone, slow and not scalable. In order to cope with the mentioned problems and scalability imposed by growing number of base stations, the 3GPP organization, which standardizes GSM, UMTS and LTE, has proposed a solution named SON (Self-Organizing Network) [46], in which certain management processes of RAN are fully automated. SON functions have been originally defined in 3GPP Release 8, which describes the LTE network, but it is also expected that they will evolve and also retrofit the previous generation, i.e., UMTS. At present 3GPP has only provided a list of SON functions, and so far the way of their implementation is manufacturer dependent. There is no doubt, that these functions require adaptive control, and that the cognitive techniques can be nicely used for SON.

SON Functions and Cognitive Techniques. All SON functions can be divided into three following groups: self-configuration functions, self-optimization functions and self-healing functions. The main goal of self-configuration functions is to obtain the Plug-and-Play base station deployment. They are: Configuration of Initial Radio Transmission Parameters, Neighbor Relation Management, Automatic Connectivity Management, Self-testing, and Automatic Inventory.

Automatic Configuration of Initial Radio Transmission Parameters lies on individual self-tuning of base station parameters, to provide proper coverage and minimize interferences among base stations. Cognitive approach to this problem has been already used. The RL combined with a multi-agent implementation has been proposed to address the problem of Inter-Cell Interference Coordination in the downlink channel [17]. In the paper, each base station is considered as an agent, which dynamically changes its transmit power to control the interference with the neighbouring cells. In the described approach FIS rules were used to handle the continuous input and output space. Moreover, an improved effective initialization procedure was provided to overcome slow convergence the drawback of Q-learning. The RL approach based on Q-Learning framework has been also proposed to handle channel sharing between small cells deployed in a macro cell [41].

Automatic Neighbour Relation Management concerns the updates of neighbour cell relationships to facilitate easy handovers between base stations. It is necessary to have up to date neighbour list to avoid dropped calls, failed handovers and QoS degradation. The manual update of neighbour relationships become not scalable and error prone, therefore it is automated in SON. The operation is very important for seamless RAN enhancement. However, it does not require cognitive approaches.

Automatic Connectivity Management lies on connectivity automation of initially deployed base station. The cognitive approach is not required in this case.

Self-testing is a process used for verification of proper functioning of base station modules. The cognitive techniques can be applied for detection of anomalies. However, the base stations internal architecture is manufacturer dependent, therefore such algorithms can only be deployed by the equipment manufacturer.

Automatic Inventory collects information about base station internals (hardware boards and software components). No cognitive techniques are applicable here.

The self-optimization SON functions are: Mobility Robustness Optimisation, Mobility Load Balancing and Traffic Steering, Energy Saving, Coverage and Capacity Optimisation, and Random Access Channel Optimisation.

- Handover (HO) is one of the key procedures for ensuring that users can move freely through the network while staying connected and being offered appropriate service quality. It is important that HO procedure happens as timely and seamlessly as possible. For quality of the handover process parameters like Radio Link Failure, Handover Failure and Handover Ping-Pong are used. The aim of **Mobility Robustness Optimisation** is to tune simultaneously handover parameters, like time-to-trigger or handover hysteresis, on the basis of previously mentioned handover performance indicators, in order to achieve optimal HO decision. Cognitive techniques are applicable to solve the problem.
- Due to the users mobility, the load of different BSs is rarely uniform. As a result of limited capacity of the network, the potential congestion may occur, which might degrade users QoS. To cope with this problem, the **Mobility Load Balancing and Traffic Steering** mechanism has been proposed. This mechanism uses information about relative load of neighbouring cells in order to slowdown or speedup HO. In [33] a distributed Q-Learning approach is used to dynamically select the HO parameters according to BSs load.
- The main purpose of **Energy Saving** lies on seamlessly turning off and on some radio nodes (BSs), sectors carriers or nodes internal blocks, to reduce the power consumption when the network load is very low (typically at night). The decision about switching an element off/on can be based on cognitive techniques. The RL algorithm, that is a model free, can be applied as in [13]. In the proposed approach each BS collects monitoring data and sends periodic report to its network cognitive manager. The manager takes the decision about switching off or on a particular BS, on the base of configuration, network topology, policy parameters and previous decision history. Cognitive techniques are appropriate one to handle this problem. However, improper behaviour of an algorithm could lead to subsequent turning off the recently turned on BS, and to QoS degradation of active data streams.
- The BS configuration parameters initially set, next can be dynamically modified by the **Network Coverage and Capacity Optimisation** function. The decision about changing BS parameters (like transmission power, antenna tilt, etc.) can be triggered by the quality reports obtained from users. The reports include such parameters like signal-to-noise ratio, timing advance or radio link failures. The improvement procedure can be based on cognitive techniques, which can use reinforcement learning. The classification of quality perceived by users can apply unsupervised learning, to reduce complexity (the input space dimension).
- In LTE mobile networks, Random Access Channel (RACH) is an uplink channel, used for initial access of a user. This is (according to the name) a random access procedure with possible conflicts. The performance of RACH procedure is dependent on network configuration and load. Measurements of RACH performance (overload), cell load and number of users, can be used in SON for **RACH Optimization** function, by the use of a cognitive technique.

The self-healing SON functions are: Cell Degradation Monitoring and Management, and Cell Outage Compensation.

Cell Degradation Monitoring and Management lies on the detection of a fault, by analysing some monitoring data available at each BS, collected from neighbouring

BSs or from users. Once a failure is detected an alarm is triggered, and if possible a healing action is taken. There are two important problems related with this function. The first one is how to discover that the measured state is degraded. The second is how to heal the network. Both problems can be nicely solved using cognitive techniques. The first one is in fact anomaly detection problem, which can be solved by unsupervised learning, whereas the second one is much complex because the healing action may involve multiple devices and mechanisms.

Cell Outage Compensation lies on the compensation of outage of a cell. The healing process has to be fast and the compensation must come from the neighbouring cells, what has to be achieved by increasing the neighbouring cells power, adjusting the antenna tilt to cover the area of the failed one. This procedure is similar to procedures of Coverage and Capacity Optimisation or Energy Saving function.

Coordination of SON Functions. The described SON functions, until recently, have been treated separately. However, in real networks they have to operate simultaneously in order to optimise the overall RAN performance. With the increasing number of SON functions, the probability of conflicts and dependences between them increases.

Such conflicts can occur if two individual SON functions aiming at different goals (e.g., Cell Outage Management and Energy Saving) try to modify the same parameter (e.g., the BS transmit power). There is also a conflict, if the modification of a triggering parameter by one SON function has negative impact on other SON functions. These conflicts may cause that the entire system will operate far from the optimal state, and the network performance and users satisfaction will be degraded. An overview of conflicts between different SON functions has been provided in the SOCRATES project [42].

As more and more SON functions are being standardized, the coordination between them is becoming more important in order to prevent and resolve unexpected and undesirable network behaviour. The coordination has to impact all individual SON functions, and at the same time to enable the operator supervising the overall SON system, and to apply operators preferences (policies). With the increasing number of SON functions, the input data space and output data space become large and multi-dimensional. The result is that mapping from input and output parameters to the optimisation objectives is very difficult to define. Moreover, the information about the input may be incomplete or not up to date (due to transmission delays, faults and dynamic changes of network state), which can cause negative impact to the decision-making process. Actually there is an on-going work on the general solution for the coordination problem, but there are no yet presented results. There are some approaches for simultaneous optimization of several SON functions. An integrated approach for two SON functions implementation, namely handover optimisation and load balancing combined with admission control is presented in [45]. In the paper the Fuzzy Q-Learning Control (FQLC) algorithm was combined with a heuristic algorithm. Both algorithms operate in different time scales.

3.2 Application of Cognitive Techniques for Management of Fixed Networks

As the number of nodes and the size of the network increase, and functionality of the devices grows, the centralized management becomes inappropriate. Hence distribution of management functions as a solution is proposed, as well as adaptation of management decisions to network status. A new approach is based on a continuous feedback from the network after making management decisions. The concept has been originally proposed by IBM for autonomic management of IT resources and described as the Monitor-Analyse-Plan-Execute (MAPE) model [1].

Using MAPE, several architectures have been proposed for network management and labelled as Autonomic Network Management (ANM). ANM includes fault handling, performance optimization (which includes energy saving approaches, configuration or dynamic reconfiguration of devices), and management of security. In general the ANM operations should concern all OSI ISO [Open System Interconnection defined by International Organization for Standardization] layers. During last six years a number of research projects has been launched, with the aim to find efficient solutions for automatic or even autonomic management of future networks. Below, we mention those, which used cognitive approaches in different contexts of network management.

The FOCALE autonomic architecture [43] defines maintenance and reconfiguration loops and contains among other reasoning and learning blocks. Data retrieved by sensors are normalized and analysed. If the managed resource follows expected states (defined by its model) the system continues using the maintenance loop. If a problem is detected, then the reconfiguration loop is used. The Autonomic Internet (AutoI) project [6] has aimed to design and develop a self-managing virtual resource overlay for heterogeneous networks. The overlay supports service mobility, QoS, and reliability. It used ontology-based information and data models, providing fast and guaranteed service delivery.

In the framework of the EFIPSANS (European 7th Framework Project), the Generic Autonomic Network Architecture (GANA) has been developed. Next ETSI has adapted and extended GANA as a Reference Model for autonomic network engineering, cognitive networking and self-management [11]. The model defines control loops for: protocol, function, node, and network levels. At every level a rich set of decision elements operates. Every decision element acts in a control loop. Algorithms are no subjects of GANA standardization. The BIONETS project [9] has aimed to find out paradigms taken from biology, physics and social science, which can be applicable for autonomic networks and services. In the project not only techniques for network management were analysed but also those, which are applicable for network control, e.g. routing, forwarding in Delay Tolerant Networks, and congestion control. In that context some cognitive techniques have been analysed [4].

The UniverSelf project [47] has aimed to design of a Unified Management Framework (UMF) for wireless and wireline networks and services. UMF has been intended to be equipped with all functions required to achieve complete self-management of networks. There are some UMF elements with learning abilities and knowledge based reasoning. The COMMUNE projects [13] goal is to cope with network management under uncertainty using cognitive techniques. The project consortium published many use cases of application of cognitive techniques to FTTH (Fiber to the Home), LTE, M2M (Machine to Machine) networks, as well as to video streaming.

3.3 Application of Cognitive Techniques to SDN

As it was outlined, in some ANM approaches the cognitive techniques are already applied. Unfortunately none of them has a commercial deployment yet. Moreover, there is no common architecture framework, for easy implementation of different cognitive techniques.

Software-Defined Networking has recently become the most important network research area, and is seen as the largest revolution in IP networks. At the present most of SDN solutions are based on the OpenFlow protocol [32], defined by Open Networking Foundation [34]. The fundamental idea of SDN networks lies on the separation of the control plane, which decides where to forward the incoming data packets/flows, and forwarding plane, which delivers the traffic to the destination. The control plane is logically centralized, and in small networks only one controller takes decisions related to all control operations. The data plane in the ONF model is composed of relatively simple switches, which main role is to forward flows according to controller decision. An important property of this approach is high-level programmability of the controller, which functions are not predefined, and are relatively simple. An OpenFlow switch can compare and alter IP packet headers, according to the controller instructions. The programmability of SDN makes the network to be application-aware, and increases the customizability and flexibility of deployment.

Following the ONF specifications [35], the applications plane proves to be a gold mine in integrating cognition, as several types of applicaitons will be needed, handling from security concerns, to performance and sustainable abilities; not to mention business specific applications. Providing efficient answers in such a dynamic environment requires the ability of learning from the previous observed network behavior and provide solutions accordingly; while having a clear separation between the SDN infrastructure management and the actual network management.

The above-mentioned programmability requires many algorithms to support network operations. The SDN controller performs many basic and also advanced tasks. The tasks include network topology discovery, network monitoring, creating of network connectivity matrix, and admission control of the user traffic. The SDN controller should perform centralized traffic engineering operations, which provide load balancing taking into account network state, achieving that way optimal usage of network resources and maximizing users satisfaction.

The controller should also perform self-management of the network, handle network topology changes and failures, and last but not least should provide basic detection of DoS security attacks. All the above mentioned operations should fulfil operator policy requirements.

There are already some approaches, which use autonomic or cognitive techniques to solve SDN problems. For example Kim, in his doctoral thesis [27], proposed an autonomic network management architecture called CogMan for SDN. The proposed approach provides reactive, deliberative, and reflective loops, where the reflective loop is intended to use learning algorithms for reasoning and for influencing the planning and decisions performed in the deliberative loop.

Centralized Traffic Engineering in SDN. The traffic engineering in SDN may benefit from the global network view. It should collect data about the network state from switches, signaled users traffic demands, and react to operator policies. With such information the Central Traffic Engineering (CTE) engine determines path assignments for individual traffic flows or group of flows, taking into account the forwarding class, the virtual price of the path and load distribution in the network. Flow level operations enable efficient dissemination of the traffic in the whole network. CTE has to provide fairness for users of the same classes of network services. In case of network topology change or failure, CTE should react quickly and recalculate relevant paths. In order to avoid the overall congestion of the network, CTE can use the admission control mechanism. For efficient usage of available network resources, CTE should apply dynamic load balancing; which should use information about network topology, as well as load of all network nodes and links. The forwarding rules should concern traffic with or without QoS guarantees, using traffic descriptors signaled by users or deduced from network observations.

CTE is a complex problem that has to be solved in real-time (the response time in range of 100 ms) for many thousands of flows. CTE algorithms should be adaptive, i.e., they should monitor some network parameters and take some actions according to their values (a feedback based approach). In relatively simple cases such algorithms can be defined a priori. But in many complex cases the proper design of algorithms requires a lot of efforts and the knowledge about the underlying mechanisms and their impact on the network behaviour. The cognitive approach is seen as a potential solution aiming to solve this problem. In such an approach, appropriate algorithms with learning abilities have to be used to achieve the predefined goals. Due to learning abilities there is no need to analyse all possible situations in fact the cognitive algorithm will adapt according to the specific network condition. The complexity of CTE suggests the usage of multiple cognitive algorithms, which have to be coordinated (see 3.1.2). So far there is none cognitive approach to traffic engineering in SDN.

DoS Attack Detection. SDN network provides the separation between the control plane and the data plane. As a result, the network offers more programmability, flexibility and customizability. The controller, with the central view, offers deeper level of granularity to packet analysis, better global network monitoring. Therefore, SDN paradigm can enhance the security level of the network. In an OpenFlow network, each switch has forwarding tables, which contain the rules for incoming packets/flows. However, in the case of reactive flow handling, the switch does not know the rule for an incoming flow, therefore a request will be directed to the controller to obtain flow rules table entry. If a network node is compromised, and constantly sends requests, then the controller will be a congestion point in the network. In case the controller is compromised, the attacker can generate fake flow controls to the network nodes, containing useless forwarding rules and overflow switches forwarding tables, or directs the flows to wrong destinations. Such situations are called DoS attack, where either controller or switches are blocked.

There is a significant need for DoS attack detection and protection in SDN. A cognitive approach that analyses traffic statistical information and different types of alarms, could detect any unusual events (i.e. abnormal traffic patterns appeared in the network). Moreover, by learning from experience, when there is a high volume of packet/flow sending in the network, the cognitive approach can detect more efficiently what is abnormal.

4 Conclusions

The flexible management and control of communication networks is an extremely important topic. Service providers, network operators, and device manufacturers look for new solutions, which enable to overcome limitations of today technologies in terms of flexibility and easiness of deployment of new services, with reduced cost of operations. Some of the mentioned in the paper technologies like SON, ANM or SDN come with such promise, however they are still not mature yet. A key role in the concepts will perform algorithms, which will drive the whole network behaviour. The described in the paper technologies are programmable not only by the manufacturer but also by operators and even by end-users. It creates a new market for software applications. In our opinion the cognitive techniques can be nicely applicable to solve many control and network management problems. The paper is an attempt to indicate in which networking areas such techniques can be applied. However the existing number of possible mechanisms, which can profit from these techniques, is so huge that selection of appropriate cognitive algorithm starts to be challenging. As it has been noticed there are already some attempts to use cognitive techniques for network control and management.

Acknowledgment. This publication is based in parts on work performed in the framework of the CoSDN project, INTER/POLLUX/12/4434480, funded by NCBiR, Poland and the Fonds National de la Recherche, Luxembourg.

References

[1] An architectural blueprint for autonomic computing. Tech. rep., IBM (June 2005)

[2] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7(1), 39–59 (1994)

[3] Agoulmine, N.: Autonomic Network Management Principles: From Concepts to Applications. Elsevier Science (2010)

[4] Altman, E., Dini, P., Miorandi, D.: Paradigms for biologically- inspired autonomic networks and services the bionets project ebook. eBook (BIONETS project EU-IST-FET-SAC-FP6-027748 project deliverable D0.2.3 (2010), www.bionets.eu

[5] Atlas, A., Nadeau, T., Ward, D.: Interface to the Routing System Framework. draft-ward-irs-framework-00 (July 2012),
http://tools.ietf.org/html/draft-ward-irs-framework-00

[6] Autonomic Internet Project, http://ist-autoi.eu/

[7] Bäck, T., Fogel, D., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing Ltd., Bristol and Oxford University Press, New York (1997)

[8] Beyer, H.G., Schwefel, H.P.: Evolution strategies - a comprehensive introduction. Natural Computing 1(1), 3–52 (2002)

[9] BIONETS, http://www.bionets.eu/

[10] Cabaj, K.: Frequent events and epochs in data stream. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 475–484. Springer, Heidelberg (2007)

[11] Chaparadza, R.: Requirements for a generic autonomic network architecture (gana), suitable for standardizable autonomic behavior specifications for diverse networking environments. Annual Review of Communications 61 (2008)

[12] Coello Coello, C.A.: List of references on evolutionary multiobjective optimization (1999), http://www.lania.mx/~ccoello/EMOO/EMOObib.html

[13] COMMUNE Celtic Project, http://projects.celtic-initiative.org/commune

[14] Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20(3), 273–297 (1995)

[15] COSDN, http://secan-lab.uni.lu/index.php/projects/cosdn

[16] Del Moral, P., Tantar, A.-A., Tantar, E.: On the foundations and the applications of evolutionary computing. In: Tantar, E., Tantar, A.-A., Bouvry, P., Del Moral, P., Legrand, P., Coello Coello, C.A., Schütze, O. (eds.) EVOLVE- A bridge between Probability, Set Oriented Numerics and Evolutionary Computation. SCI, vol. 447, pp. 3–89. Springer, Heidelberg (2013)

[17] Dirani, M., Altman, Z.: A cooperative reinforcement learning approach for inter-cell interference coordination in ofdma cellular networks. In: WiOpt, pp. 170–176. IEEE (2010)

[18] European Telecommunications Standard Institute (ETSI), http://www.etsi.org

[19] Goldberg, D., Korb, B., Deb, K.: Messy Genetic Algorithms: Motivation, Analysis, and First Results. Complex Systems 3(5), 493–530 (1989)

[20] Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: International Conference on Evolutionary Computation, pp. 312–317 (1996)

[21] Hartigan, J., Wong, M.: Algorithm AS 136: A K-means clustering algorithm. In: Applied Statistics, pp. 100–108 (1979)

[22] Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge (1992)

[23] Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America 79(8), 2554–2558 (1982)

[24] Jensen, F.V.: Introduction to Bayesian Networks, 1st edn. Springer-Verlag New York, Inc., Secaucus (1996)

[25] Jolliffe, I.: Principal Component Analysis, 2nd edn. Springer (October 2002)

[26] Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man, and Cybernetics 15(4), 580–585 (1985)

[27] Kim, S.: Cognitive Model-Based Autonomic Fault Management in SDN. Ph.D. thesis, Pohang University of Science and Technology (2013)

[28] Kohonen, T.: Self-organized Formation of Topologically Correct Feature Maps. In: Neurocomputing: Foundations of Research, pp. 509–521. MIT Press, Cambridge (1988)

[29] Langdon, W.B., Poli, R.: Foundations of genetic programming. Springer (2002)

[30] Larrañaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer, Boston (2002)

[31] Mannie, E.: Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945 (Proposed Standard) (October 2004), http://www.ietf.org/rfc/rfc3945.txt

[32] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: Enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38(2), 69–74 (2008)

[33] Mwanje, S., Mitschele-Thiel, A.: A q-learning strategy for lte mobility load balancing. In: Proceedings of the 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2013), London, UK (September 2013)

[34] Open Networking Foundation, http://opennetworking.org

[35] Open Networking Foundation: SDN architecture overview. ONF White paper (December 2013)

[36] Pawlak, Z.: Rough set theory and its applications to data analysis. Cybernetics and Systems 29(7), 661–688 (1998)

[37] Rabiner, L., Juang, B.: An introduction to hidden Markov models. IEEE ASSP Magazine 3(1), 4–16 (2003)

[38] Rao, S., Shantha, C.: Numerical Methods: With Programs in BASIC, FORTRAN, Pascal and C++. Universities Press, India (2004)

[39] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)

[40] Rust, J.P.: Structural estimation of markov decision processes. In: Engle, R.F., McFadden, D. (eds.) Handbook of Econometrics, 1st edn., vol. 4, ch. 51, pp. 3081–3143. Elsevier (1986)

[41] Simsek, M., Czylwik, A., Galindo-Serrano, A., Giupponi, L.: Improved decentralized q-learning algorithm for interference reduction in lte-femtocells. In: Wireless Advanced (WiAd), pp. 138–143 (June 2011)

[42] SOCRATES, http://www.fp7-socrates.eu

[43] Strassner, J., Agoulmine, N., Lehtihet, E.: FOCALE: A Novel Autonomic Networking Architecture (2006)

[44] Sutton, R., Barto, A.: Reinforcement learning: An introduction. MIT Press, Cambridge (1998)

[45] Truong, K., Kuklinski, S.: Joint implementation of several lte-son functions. In: GLOBECOM Workshops, Atlanta, USA (December 2013)

[46] Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements. 3GPP TS 32.500 (version 8.0.0, Release 8) (December 2008)

[47] UniverSelf, http://www.univerself-project.eu

[48] Walker, M.: Introduction to Genetic Programming (October 2001), http://www.cs.montana.edu/~bwall/cs580/introduction_to_gp.pdf

[49] Zimmermann, H.J.: Fuzzy Set Theory and its Applications. Springer (2001)

Part IV

Complex Networks and Landscape Analysis

Pareto Landscapes Analyses via Graph-Based Modeling for Interactive Decision-Making

Ofer M. Shir[1,2,*], Shahar Chen[1,3,*], David Amid[1], Oded Margalit[1],
Michael Masin[1], Ateret Anaby-Tavor[1], and David Boaz[1]

[1] IBM Research, Haifa, Israel
[2] Computer Science Department, Tel-Hai College, Upper Galilee, Israel
[3] Computer Science Department, Technion - Israel Institute of Technology

Abstract. We consider two complementary tasks for consuming optimization results of a given multiobjective problem by decision-makers. The underpinning in both exploratory tasks is analyzing Pareto landscapes, and we propose in both cases discrete graph-based reductions. Firstly, we introduce interactive navigation from a given suboptimal reference solution to Pareto efficient solution-points. The proposed traversal mechanism is based upon landscape improvement-transitions from the reference towards Pareto-dominating solutions in a *baby-steps fashion* – accepting relatively small variations in the design-space. The Efficient Frontier and the archive of Pareto suboptimal points are to be obtained by population-based multiobjective solvers, such as Evolutionary Multiobjective Algorithms. Secondly, we propose a framework for automatically recommending a preferable subset of points belonging to the Frontier that accounts for the decision-maker's tendencies. We devise a line of action that activates one of two approaches: either recommending the top *offensive team* – the gain-prone subset of points, or the top *defensive team* – the loss-averse subset of points. We describe the entire recommendation process and formulate mixed-integer linear programs for solving its combinatorial graph-based problems.

Keywords: Pareto landscapes, multi-criterion decision-making, design-space, interactive recommender systems, prospect theory, graph traversals, vertex covering, submodular functions.

1 Introduction

The goal of multiobjective optimization is to output the Efficient Frontier and the Pareto Optimal Set, whose points are mathematically indifferent with respect to each other. Attaining the Efficient Frontier of a multiobjective optimization problem can be either treated by means of algorithms utilizing mathematical programming solvers, e.g., the so-called Diversity Maximization Approach [1] employing specific solvers, or approximated by population-based heuristics, such as Evolutionary Multiobjective Optimization Algorithms [2]. The selection phase, entitled Multi-Criterion Decision Making (MCDM) [3], is to be subjectively driven by the human decision-maker (DM) based

[*] The first two authors contributed equally to this work.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_8, © Springer International Publishing Switzerland 2014

Nomenclature

Term	Description	Notation
model	multiobjective optimization model	$\mathcal{M}$
Efficient Frontier	non-dominated points in the m-dimensional objective space	$\mathcal{F}$
Pareto Optimal Set	pre-images of the Frontier in the search space	$\mathcal{X}$
archive	history of solution-points obtained during optimization	$\mathcal{A}$
solution	a feasible solution to the given multiobjective problem	ψ
reference	a solution provided by the user to become an initial point	ψ_0
transition	an ordered pair of solution-points a and b	$(a \rightsquigarrow b)$
path	a sequence of transitions from the reference to the Frontier	$p,\ \mathcal{P}[:,:]$
metric	symmetric distance metric amongst solution-points	Δ
hop limit	maximally acceptable distance for transitions	δ_{max}
preference	a preference relation between two solution-points	$\preceq_{\text{pref}}$
graph	a graph with vertices V and directed edges E	$\mathcal{G} = \{V, E\}$
edge	a directed edge between node u to node v with weight ω	$(u \rightsquigarrow v,\ \omega)$

upon their preferences. This phase may involve exploration of the Frontier, and eventually, the challenge in selecting a solution is to account for gains and losses while adhering to the personal preferences. The current work is concerned with devising automated exploration recipes in two directions: (i) utilizing a *traversal mechanism* to discover (possibly suboptimal) points that may be of interest because of their design-space specification, and (ii) identifying recommended subsets of solution-points that meet the DM's tendencies. We propose here a common ground to treat both exploration tasks, that is, by reducing the challenge of landscape analyses to discrete graph-based formulations. While the traversal challenge is reduced to a *shortest path* problem on the archive of suboptimal solutions, the recommendation challenge is reduced to a *vertex covering problem* of an outranking graph. In the latter we show how to transform the analysis problem into mixed-integer linear programs as well as a formulation of maximizing a *submodular function* over the graph vertices. The Nomenclature for this paper is outlined in the enclosed table.

Related Work. In essence, work related to the considered tasks lies in the areas of MCDM and interactive recommender systems. Preference elicitation and derivation of expected utility functions is rooted in the Multi-Attribute Utility Theory [4]. Analytical Hierarchy Process [5] introduced a questioning protocol for manual decision-making, based upon pairwise comparisons, and derived a calculation procedure featuring matrix algebra. The ELECTRE family of methods [6], on the other hand, defined a preference relation amongst solution-points, and proposed a procedure to construct a preference graph – these methods eventually recommend to the DM the solution-points that lie in this graph's *kernel*. Various approaches for reducing the cardinality of the Efficient Frontier and computing a subset of the "most interesting" solution-points have been proposed from different angles, e.g., in *preference ordering* [7] or in *smart filtering* [8]. Interactive recommender systems that suggest candidate solution-points to the user while accounting for multiple attributes (objectives) have long been proposed, e.g., in the realm of e-Commerce [9]. These methods typically infer the user's preferences that are communicated via so-called *critiques* in order to adjust the next suggested solution. Another approach, that is somewhat related to our proposed traversal, is the so-called Pareto Navigator [10], which interactively incorporates user preferences to the course of optimization. Finally, employing a graph-based approach to navigate amongst suggested solution-points has been proposed by Hadzic and O'Sullivan [11]. Their study aimed to select a product from a catalogue (e-Commerce) without considering explicit Pareto relations, and suggesting candidate solution-points by analyzing the graph.

2 Traversing over Pareto Landscapes

In order to promote acceptance of machine-driven optimization results by DMs and gain their confidence in such candidate solutions, we present here a novel exploration framework for the multiobjective domain, which constitutes a traversal over Pareto landscapes. The primary idea behind the Pareto traversal is to offer a hopping mechanism that begins at the DM's suboptimal reference point and terminates at the Efficient Frontier or prior to that, at any point satisfactory to the DM. In short, each hop is directed towards Pareto-improvements, and at the same time is meant to constitute a *baby-step* in the **design (decision) space**, i.e., transitions are improvements in the prescribed objectives but are only relatively small variations with respect to the current candidate design. The rationale behind this constraint is the fact that human perception tends to adjust better to small changes rather than to large. At the same time, a typical limitation on human *mental resource* upon reaching a decision [12] poses a tradeoff between the need for baby-step adjustments to the available mental resource (time and energy to decide).

The strict Pareto domination relation can be relaxed by means of a generalized preference relation between two solution-points. While we keep our formal definition of the Efficient Frontier as the output of the optimization process, we would like to consider a relaxed preference relationship amongst solution-points in the archive. Examples of such relations are described in [13], entitled therein (ε, λ)-dominance or the preference-elicited $\preceq_{\Theta}$ relation. In what follows, we shall denote a preference relation in this framework as $\preceq_{\text{pref}}$, and it is assumed to be prescribed by the user. The careful reader would note that this relation is not necessarily a partial order, and a preference graph that is based upon $\preceq_{\text{pref}}$-relations may possess cycles.

2.1 Proposed Method I: Batch Navigation

A successful Pareto optimization process yields the Efficient Frontier as well its Pareto Optimal Set, but may support in parallel the **archiving** of all its intermediate feasible solutions along this process. The obtained archive, whose non-dominated set is the Efficient Frontier, is likely to contain a vast majority of dominated solutions with diverse ranks. **This archive is of particular interest to the current study, since it encompasses the potential to bridge between a reference solution to Pareto optimal solutions by means of a guided hopping mechanism**. Our proposed Pareto navigation system offers 3 optional tracks of guidance to the DM. In all cases, a symmetric distance metric is provided as input:

1. **Closest:** given a reference solution and an archive, return a Pareto efficient solution that is closest to the reference point in the scope of the design space.
2. **Shortest:** given a reference solution, an archive, a preference relation $\preceq_{\text{pref}}$, and the hop limit $\delta_{\max}$, construct the directed graph of all possible pathways, and return the shortest path from the reference to the Frontier. The shortest path can be

```
shortestPathToFrontier (reference $\psi_0$, Frontier $\mathcal{F}$, Archive $\mathcal{A}$,
                        metric $\Delta$, preference $\preceq_{\text{pref}}$, lim $\delta_{\max}$)
$\mathcal{G} \longleftarrow$ constructTraversalGraph($\mathcal{A}, \Delta, \preceq_{\text{pref}}, \delta_{\max}$)
$\{\mathcal{D}, \mathcal{P}\} \longleftarrow \mathcal{G}$.Dijkstra($\psi_0, \mathcal{F}$)
idx $\longleftarrow \arg\min(\mathcal{D})$   /* minimal accumulated distance to the Frontier */
return $\{\mathcal{P}[\text{idx}], \text{idx}\}$
```

either defined as the shortest sequence of hops to the Frontier (in terms of number of moves), or as the smallest accumulated distance to the Frontier. The design distance between each pair of consecutive solutions is bounded by $\delta_{\max}$. See shortestPathToFrontier for the pseudo-code.

3. **Classic:** given a reference solution, an archive, a preference relation, a maximal distance, and maximal number of suggested solutions, iteratively return a set of dominating solutions, close enough to the reference (bounded by $\delta_{\max}$ in the design space), lying on paths to the Frontier and of which the DM is requested to select. See traverseToFrontier for the pseudo-code.

We discuss in what follows the various components of the **Classic** traversal.

Shortest Path Calculations. We view the archive of the optimization process $\mathcal{A}$ as a *directed graph*: each solution-point constitutes a vertex, and there is an edge from solution-point a to solution-point b if and only if $b \preceq_{\text{pref}} a$ holds and the design distance between a and b does not exceed $\delta_{\max}$. For instance, if $\preceq_{\text{pref}}$ is taken to be the standard Pareto domination – and if $\Delta(a, b) \leq \delta_{\max}$ and b dominates a both hold – then the graph will possess an edge $(a \rightsquigarrow b,\ \Delta(a, b))$. See constructTraversalGraph for the pseudo-code. By the current construction, a valid sequence of solutions to the Frontier is a valid path from the source vertex (ψ_0, the reference solution provided by the user) to one of the Frontier vertices, i.e., a solution-point in $\mathcal{F}$. In practice, we consider Dijkstra's shortest path algorithm [14], utilized on a directed graph $\mathcal{G}$, which receives a source ψ_0 and the Efficient Frontier vertices $\mathcal{F}$. Its implementation returns the shortest paths from the source to each of the vertices in $\mathcal{F}$, with both time and space complexity of $\mathcal{O}\left(|\mathcal{A}|^2\right)$. Its output, denoted in our notation as $\mathcal{G}$.Dijkstra, comprises the calculated distances $\mathcal{D}$ and the description of the paths $\mathcal{P}$. As mentioned earlier, $\mathcal{G}$ may possess cycles depending upon the definition of $\preceq_{\text{pref}}$, e.g., a cycle of indifferent solution-points when $\preceq_{\text{pref}}$ is a weak Pareto-dominance relation. In such cases, Dijkstra's algorithm is still guaranteed to calculate the shortest paths to $\mathcal{F}$, but a feature to eliminate revisiting nodes iteration-wise is then needed (since Dijkstra is executed independently in each iteration), e.g., by means of Tabu-list elimination. A discussion on paths calculation efficiency in large-scale archives will follow.

The Interactive Hopping Mechanism. Each iteration is meant to hold a specific solution, ψ_t, starting with the original reference ψ_0 at the beginning. At iteration t, the proposed method presents several candidate hops, and more specifically, it presents a limited number of solutions ($N_{\max}$ at most) that meet all the following **3 requirements**:

```
constructTraversalGraph (Archive $\mathcal{A}$, metric $\Delta$, preference $\preceq_{\text{pref}}$, lim $\delta_{\max}$)
$V \longleftarrow \mathcal{A},\ E \longleftarrow \emptyset$
for $i = 1 \ldots |\mathcal{A}|$ do
    for $j = 1 \ldots |\mathcal{A}|$ do
        if $\mathcal{A}[i] \preceq_{\text{pref}} \mathcal{A}[j] \ \wedge\ \Delta(i,j) \leq \delta_{\max}$ then
            $E \longleftarrow E \cup (j \rightsquigarrow i,\ \omega = \Delta(i,j))$
        end if
    end for
end for
return $\mathcal{G} = \{V, E\}$
```

```
traverseToFrontier (reference $\psi_0$, Frontier $\mathcal{F}$, Archive $\mathcal{A}$,
                    metric $\Delta$, preference $\preceq_{\text{pref}}$, lim $\delta_{\max}$, suggestions $N_{\max}$)
 1: $\mathcal{G} \longleftarrow$constructTraversalGraph($\mathcal{A}, \Delta, \preceq_{\text{pref}}, \delta_{\max}$)
 2: $t \longleftarrow 0,\ p \longleftarrow \emptyset$
 3: repeat
 4:     $\{\mathcal{D},\ \mathcal{P}\} \longleftarrow \mathcal{G}$.Dijkstra($\psi_t, \mathcal{F}$)    /* obtain distances and paths to $\mathcal{F}$ */
 5:     $\mathcal{I} \longleftarrow$sort($\mathcal{D}$, 'ascend')    /* hold the post-sorting permutation indices */
 6:     $\mathcal{H}_{\mathcal{G}}(\psi_t) \longleftarrow \{v \mid v \in \mathcal{P}[1, \mathcal{I}(1 : N_{\max})]\}$
 7:     $\boxed{\psi_{t+1} \longleftarrow \texttt{getDMSelection}(\mathcal{H}_{\mathcal{G}}(\psi_t))}$ /* get input from the user */
 8:     $p[t] \longleftarrow (\psi_t \rightsquigarrow \psi_{t+1})$
 9:     $t \longleftarrow t + 1$
10: until terminatedByDM()
11: print($p$)
12: return $\psi_t$
```

1. Satisfying the preference relation $\preceq_{\text{pref}}$ with regard to ψ_t (objective-wise)
2. Being within $\delta_{\max}$ from ψ_t (design-wise with respect to Δ)
3. Frontier-reachable: on a pathway of valid hops (i.e., via nodes that satisfy (1)+(2)) to the Efficient Frontier

The peak of each iteration is reached when the DM is asked by the machine to select the next hop within the proposed set, which we denote as $\mathcal{H}_{\mathcal{G}}$; this step is entitled `getDMSelection`: $\psi_{t+1} \longleftarrow$ `getDMSelection`$(\mathcal{H}_{\mathcal{G}}(\psi_t))$. Note that ψ_t may alternatively be defined as a set, rather than an individual solution, and the procedure can be adapted accordingly. A set perspective may suit certain DMs that are capable of simultaneously consuming multiple pathways.

Selection Criteria. When calculating the shortest paths from the current solution-point ψ_t to the Frontier $\mathcal{F}$, as in `traverseToFrontier`, attention must be paid to the selection criterion amongst the paths. This is of particular relevance in scenarios where a large number of pathways exist, and the interaction with the DM should comprise a relatively small number of representative candidates. The default selection criterion (line 6 of `traverseToFrontier`) considers the minimization over the accumulated design space distance between ψ_t to $\mathcal{F}$ and the selection of the top $N_{\max}$ minimizers. While this criterion constitutes the primary motivation behind this newly proposed technique, there could be alternative criteria, or secondary criteria that would follow it. Possible ideas may comprise minimizing the total number of hops to the Frontier (i.e., considering the shortest path subject to graph's edge lengths uniformly set to 1), maximizing the overall improvements in all objectives (e.g., considering a global utility of all objective functions and accounting for its largest ascend), maximizing the *design diversity* in the target solution-points on the Frontier, or maximizing the objective diversity amongst the $N_{\max}$ selected points. The careful reader may foresee scenarios in which no paths exist per certain $\delta_{\max}$ values. Alternatively, the possibility of paths that comprise too many hops also exists. In both cases, we propose an interactive approach that may either iteratively adapt $\delta_{\max}$ based on feedback from the user, self-adapt it according to a heuristic, or hybridize the two approaches. At the same time, scenarios in which the local neighborhood of the current solution-point possesses an excessive number of candidates may arise. In these scenarios, we propose to apply secondary selection criteria in order to trim its size.

Space-Complexity in Large-Scale Archives. A practical problem arises when the graph cardinality becomes excessively large (e.g., order of 20000 nodes). In order to address this issue, we aimed to exploit the tradeoff between memory and run-time and implemented a lighter version of Dijkstra that does not hold the entire graph in memory but rather calculates edge lengths *on-the-fly*. Consequently, the **space complexity** is reduced to $\mathcal{O}\left(|\mathcal{A}|\right)$ in comparison to $\mathcal{O}\left(|\mathcal{A}|^2\right)$ in the original algorithm. On the other hand, the graph edges are generated during the execution of the function, and therefore the graph construction is also performed more than once. Assuming the graph construction runs in t_1 seconds, and Dijkstra runs in t_2 seconds (both are $\mathcal{O}\left(|\mathcal{A}|^2\right)$), then k executions of Dijkstra-Light will take $k \cdot (t_1 + t_2)$, instead of $t_1 + k \cdot t_2$. Let $c = \frac{t_2}{t_1}$ denote the run-time ratio between graph construction and Dijkstra execution. Thus we obtain:

$$\frac{k \cdot (t_1 + t_2)}{t_1 + k \cdot t_2} = \frac{k \cdot (c+1)}{1 + k \cdot c} < \frac{k \cdot (c+1)}{k \cdot c} = 1 + \frac{1}{c}. \tag{1}$$

In practice, graph construction tends to be significantly faster than Dijkstra execution, implying a loss of at most factor-2 in run-time with our so-called Dijkstra Light. Other heuristic techniques to treat large-scale archives may comprise filtering-out the following subsets: (a) non-dominating archive solutions (solutions that do not Pareto dominate the reference solution are not reachable and thus should be excluded from the graph), (b) unreachable vertices (i.e., solutions that are not reachable within $\delta_{\max}$ steps from ψ_0), and (c) vertices that do not belong to paths ending on the Frontier $\mathcal{F}$.

An illustration of the batch navigation process is provided in Fig. 1 on a 5-objective problem by means of Parallel-Coordinates visualization.

2.2 Proposed Method II: Online Navigation

This approach does not perform any calculations in advance, but rather solves the optimization model *on-the-fly* with interactive guidance by the DM. In what follows, reference is made to the pseudo-code entitled `onlineTraversal`. Here, a multiobjective

```
onlineTraversal (model $\mathcal{M}$, reference $\psi_0$, metric $\varDelta$, lim $\delta_{\max}$)
 1: $t \longleftarrow 0$
 2: $p \longleftarrow \emptyset$
 3: repeat
 4:     $\mathcal{R} \longleftarrow$ formMO($\mathcal{M}, \psi_t, \varDelta, \delta_{\max}$)    /* formulate a local multiobjective problem */
 5:     $\mathcal{N}_{\mathcal{G}}(\psi_t) \longleftarrow$ solveLocal($\mathcal{R}$)
 6:     $\boxed{\psi_{t+1} \longleftarrow \texttt{getDMSelection}(\mathcal{N}_{\mathcal{G}}(\psi_t))}$    /* get input from the user */
 7:     $p[t] \longleftarrow (\psi_t \rightsquigarrow \psi_{t+1})$
 8:     $t \longleftarrow t + 1$
 9: until terminatedByDM()
10: print($p$)
11: return $\psi_t$
```

optimization model $\mathcal{M}$ is provided as input to the algorithm, which then dynamically formulates a variant model $\mathcal{R}$ (line 4) in each iteration and solves it (line 5) on-the-fly to obtain the local Efficient Frontier with regard to the current solution-point ψ_t. The locality of the attained solutions in the design space has to satisfy the constraints prescribed by $\mathcal{R}$, e.g., $\varDelta(\psi_t, \boldsymbol{x}) \leq \delta_{\max}$. An example for a multiobjective optimization approach that accounts for decision space constraints was reported in [15]. The selection of the decision maker is recorded (line 6) and becomes the solution-point of the consecutive iteration.

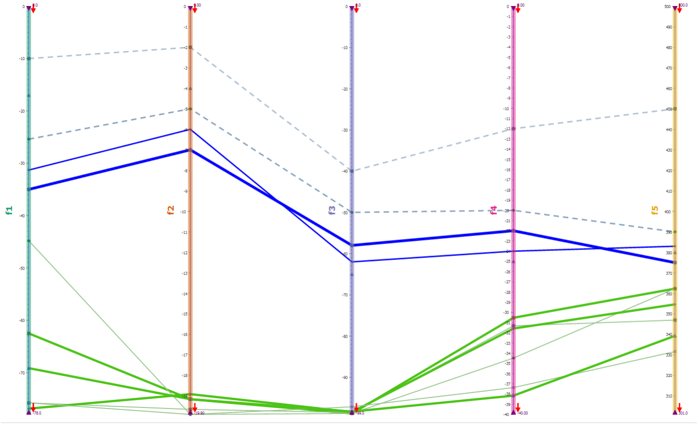


Fig. 1. A realization of the batch navigation process for a 5-objective minimization problem, depicted by means of Parallel Coordinates, with arbitrary units. The dotted lines represent the already traveresed path, whereas the solid blue lines represent two candidate solution-points being proposed to the DM at the current iteration. Furthermore, the blue solid **thick** line represents the currently examined solution-point, and the green solid **thick** lines are solution-points that may be reached from the examined option via calculated feasible pathways.

3 Automated Recommendation

In this section we are interested in formulating an automated methodology for recommending specific solution-points within the Efficient Frontier with minimal *a priori* information elicited from the DM. The Efficient Frontier is assumed to be computed in a satisfactory manner in advance and to be provided as input to the proposed process (Generate-First-Choose-Later fashion). One of the principal directions of this study is the consideration of Prospect Theory [16], whose core involves two perspectives, namely gain-prone versus loss-aversive, in human decision-making. While being gain-prone and striving to maximize profit is the economically rational perspective, it is argued that being loss-aversive and avoiding temporary losses/risk at any cost often *takes-over* human decision-making. Here, the general idea is to adopt this consideration and form two tracks of recommendation – *loss-averse* (LA) and *gain-prone* (GP) perspectives – and introduce them to recommender systems in MCDM. We coin terms, which are borrowed from American Football, and consider respectively two *teams*: the LA team, formed by the solution-points excelling in *defense*, alongside the GP team, formed by the solution-points excelling in *offense*. This partition to tracks is not revealed to the DMs, and the recommender system is to select the appropriate track according to their elicited tendencies. Given an Efficient Frontier of size N, we propose to either consider it as a whole or partition it to clusters, and treat the tasks of either *global* or *local* recommendations, respectively. The partitioning of the Efficient Frontier, by means of unsupervised clustering, is to be followed by the identification of so-called winners per each partition, yielding overall n recommended solutions. The framework is summarized by the following steps:

1. Partitioning the Efficient Frontier, $\mathcal{F} = \left\{\boldsymbol{f}^{(i)}\right\}_{i=1}^{N}$, into κ clusters of sizes $N_1, N_2, \ldots, N_\kappa$.
2. For each cluster i, constructing a **complete directed graph** with N_i vertices, each representing a solution-point. Pairwise outranking calculations obtain the degree to which a solution is preferred over another (if at all), and consequently, a weighted edge is constructed between them directed toward the outranked solution. The weight, $w_e \in [0, 1]$, quantifies the degree of preference of one solution over the other, by construction. The calculation of this degree is inspired by *Fuzzy Logic K-Optimality* [17], and certain variants of the ELECTRE family of methods [6].
3. Selecting a "good subset" of vertices for each cluster, referred to as the "winners" or the "top team".

Fig. 2 summarizes the entire recommendation process that we envision. Also, see `recommend` for the pseudo-code of the proposed recommendation recipe.

3.1 Clustering

Clustering is meant to reduce the complexity of dealing with a potentially large Efficient Frontier into treatment of its semantically-derived subsets. We propose to divide the Frontier into smaller regions and let the DM focus only on areas that they find particularly interesting. We consider each cluster as an independent set of solution-points, and recommend n_i out of N_i solutions that reside in the i^{th} cluster ($\frac{n_i}{N_i} \approx \frac{n}{N}$). A possible realization is Lloyd's k-means clustering [18].

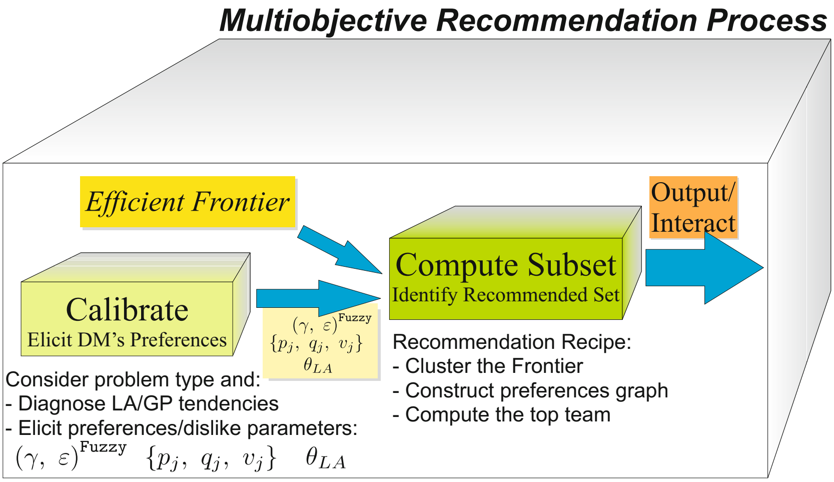

Multiobjective Recommendation Process
Efficient Frontier
Calibrate
Elicit DM's Preferences
$(\gamma,\ \varepsilon)^{\text{Fuzzy}}$
$\{p_j,\ q_j,\ v_j\}$
θ_{LA}
Compute Subset
Identify Recommended Set
Output/
Interact
Consider problem type and:
- Diagnose LA/GP tendencies
- Elicit preferences/dislike parameters:
$(\gamma,\ \varepsilon)^{\text{Fuzzy}}$ $\{p_j,\ q_j,\ v_j\}$ θ_{LA}
Recommendation Recipe:
- Cluster the Frontier
- Construct preferences graph
- Compute the top team

Fig. 2. Summary of the envisioned recommendation process. The current study focuses on the subset attainment, whereas the calibration is not addressed here.

```
recommend(Efficient Frontier $\mathcal{F}$, numClusters $\kappa$, int mode)
$\Gamma \longleftarrow$ cluster($\mathcal{F}$, $\kappa$)
for $i = 1 \ldots \kappa$ do
    $\mathcal{G}_i \longleftarrow$ calcOutrankingGraph($\mathcal{F}(\Gamma(i,:))$)
    if mode==GP then
        $\mathcal{W}_i \longleftarrow$ selectOffensiveTeam($\mathcal{G}_i$)
    else
        $\mathcal{W}_i \longleftarrow$ selectDefensiveTeam($\mathcal{G}_i$)
    end if
end for
return $\{\mathcal{W}_i\}_{i=1}^{\kappa}$    /* top teams per cluster */
```

3.2 Outranking Relations: Pairwise Comparisons

As a second step, the machine performs a series of pairwise comparisons amongst the solution-points in each cluster. Comparing two solutions a and b and determining what is the confidence level that a is preferable over b is somewhat a simpler task than conducting global prioritization over a set of solutions. Thus, we perform $\sum_{1\leq i\leq\kappa} 2\binom{N_i}{2}$ pairwise comparisons (rather than $2\binom{N}{2}$ in the global perspective, which is $\sim \kappa$ faster), and a directed weighted graph encompassing this entire information is then constructed per each cluster. In essence, an edge in this graph, directed from vertex a toward vertex b, represents the confidence level regarding the assertion ”solution-point a is favorable over solution-point b”. We consider three estimation techniques for quantifying such confidence levels: one is based upon K-Optimality and *Fuzzy Logic* [17], while the

```
calcOutrankingGraph(solutions $\mathcal{F}$)
initialize pairwise preference matrix $\Omega = (\omega_{i,j}) \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$, $\omega_{i,i} = 0$
$V \longleftarrow \mathcal{F}$
$E \longleftarrow \emptyset$
for $i = 1 \ldots |\mathcal{F}|$ do
    for $j = 1 \ldots |\mathcal{F}|$ do
        switch (mode)
        case K-OPT:
            $\omega_{i,j} \longleftarrow 1/\min_k \left(\boldsymbol{f}^{(i)} \prec_k \boldsymbol{f}^{(j)}\right)$   /* see Eq. 3 */
        case ELECT-III:
            $\omega_{i,j} \longleftarrow \sigma\left(\boldsymbol{f}^{(i)},\ \boldsymbol{f}^{(j)}\right)$   /* see Eq. 4 */
        case ELECT-IS:
            $\omega_{i,j} \longleftarrow \max_s \left(\boldsymbol{f}^{(i)} \preceq_s \boldsymbol{f}^{(j)}\right)$   /* see Eq. 5 */
        end switch
        $E \longleftarrow E \cup (i \rightsquigarrow j,\ w = \omega_{i,j})$
    end for
end for
return $\mathcal{G} = \{V, E\}$   /* complete directed graph */
```

other two stem from the so-called *ELECTRE-III* and *ELECTRE-IS* techniques [6]. See calcOutrankingGraph for the pseudo-code of the graph construction.

Preference Estimation with K-Optimality and Fuzzy Logic. Following [17], we adopt the k-dominance relation, which we shall define in what follows concerning solution-points $\boldsymbol{f}^{(1)}$ and $\boldsymbol{f}^{(2)}$ subject to vector minimization. First, consider the following functions,

$$n_b\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) \equiv \left|\left\{i \in [0\ldots m] \,\middle|\, f_i^{(1)} < f_i^{(2)}\right\}\right|$$
$$n_e\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) \equiv \left|\left\{i \in [0\ldots m] \,\middle|\, f_i^{(1)} = f_i^{(2)}\right\}\right| ,$$
$$n_w\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) \equiv \left|\left\{i \in [0\ldots m] \,\middle|\, f_i^{(1)} > f_i^{(2)}\right\}\right|$$

counting better/equal/worse coordinates in a pairwise vector comparison. Let m denote the objective-space dimensionality, then the following straightforward equality and inequality hold on any **two different solution-points residing on the Efficient Frontier**: $n_b + n_e + n_w = m, \quad 0 \leq n_b, n_e, n_w < m$. We employ *Fuzzy Logic* to relax this standard definition. When conducting pairwise comparisons, accounting for the *level of improvement* between two solutions may become an important factor. In order to encompass this aspect, the important concept of fuzziness is discussed here in the form of fuzzy membership functions. The definition of n_b, n_e and n_w can be revisited upon consideration of the fuzzy membership functions μ_b, μ_e and μ_w (note the aggregated subscripts):

$$n_{\{b,e,w\}}\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) \equiv \sum_{i=1}^{m} \mu_{\{b,e,w\}}^{(i)}\left(f_i^{(1)} - f_i^{(2)}\right)$$

The specific shape of membership is to be decided upon, e.g., by setting it to linear fuzzy membership functions. Given these $\{n_b, n_e, n_w\}$ measures, evaluated by means of the direct or fuzzy notion, we would like to form hierarchy of sub-classes of solutions within the Efficient Frontier. We state that solution-point $\boldsymbol{f}^{(1)}$ k-dominates solution-point $\boldsymbol{f}^{(2)}$, denoted as $\boldsymbol{f}^{(1)} \prec_k \boldsymbol{f}^{(2)}$, if and only if

$$\boldsymbol{f}^{(1)} \prec_k \boldsymbol{f}^{(2)} \Longleftrightarrow n_e\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) < m \wedge \frac{n_w\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right)}{n_b\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right)} \leq k \tag{2}$$

with $0 \leq k \leq 1$. Accordingly, k-optimality can be then defined: given $0 \leq k \leq 1$, $\boldsymbol{v}^*$ is the k-optimum if and only if there is no other solution $\boldsymbol{v}$, such that $\boldsymbol{v}$ k-dominates $\boldsymbol{v}^*$. Eq. 2 with $k = 0$ is essentially reduced to the formal Pareto domination relation, i.e., all the solution-points on the Efficient Frontier are 0-optimal in the strict (non-fuzzy) notion. We introduce the *degree of k-optimality* of a given solution-point $\boldsymbol{v}$ as the maximal k for which $\boldsymbol{v}$ is k-optimal. Hence, the "resistance" of a given solution increases as its degree of optimality increases, and therefore, a solution with a larger degree is more likely to be preferred over others. In order to compare amongst solution-points, we **define** the following pairwise preference function regarding solution-points a and b lying on an Efficient Frontier ($k > 0$):

$$F_{\texttt{K-OPT}}(a, b) = \begin{cases} 1/\min_k (a \prec_k b) & \text{if } \texttt{min} \text{ non-empty} \\ 0 & \text{otherwise} \end{cases} . \tag{3}$$

Preference Estimation with ELECTRE. The *ELECTRE* family of techniques [6] introduces a different approach for conducting automated pairwise comparisons. Generally speaking, ELECTRE focuses on quantifying the degree of loss/inferiority when comparing between solution-points a and b. Accordingly, as the level of loss increases, the assertion that a is preferred over b weakens. Here, the first value to be computed is the *concordance*, denoted as $c_{(q)}(a, b)$, which is defined as the fraction of objectives in which the values in solution-point a outrank the values in solution-point b with a tolerated error of q. This parameter, as others, may be utilized either with or without a coordinate subscript, indicating that it is either per-coordinate or global, respectively. At the same time, a *discordance* index $d^{(j)}_{(p,v)}$ is defined as the complementary measure to the concordance index; it is formulated for the j^{th} coordinate, and it accounts for losses between p_j to v_j. Losses beyond the latter parameter, which is entitled the *veto threshold*, would eliminate the outranking assertion. In other words, $d^{(j)}_{(p,v)} \in [0, 1]$ is proportional to the loss in $[-p_j, -v_j]$ – if the loss exceeds $-v_j$ then $d_j = 1$ and veto is applied; if the loss does not exceed $-p_j$ then $d_j = 0$. We are interested in two particular ELECTRE variants, as specified in what follows.

ELECTRE-III. Here, the score for the assertion that a outranks b is defined using a *credibility index* $\sigma(a, b)$, which quantifies the certainty of that assertion:

$$\sigma(a, b) = c(a, b) \cdot \prod_{j: d^{(j)}_{(p,v)}(a,b) > c_{(q)}(a,b)} \frac{1 - d^{(j)}_{(p,v)}(a, b)}{1 - c_{(q)}(a, b)}$$

We propose this index as a **definition** for a pairwise preference function:

$$F_{\texttt{EL-III}}\left(a,b\right)=\sigma\left(a,b\right) \tag{4}$$

ELECTRE-IS. In this variant, on the other hand, a threshold s is given, and the assertion that a outranks b is quantified by means of a boolean index, denoted as $a \preceq_s b$, and holds if and only if

$$\begin{cases} c_{(q)}\left(a,b\right)\geq s \\ \forall j \;\; d_{(p,v)}^{(j)}\geq -v+\left(v-p\right)\cdot w\left(s,c_{(q)}\left(a,b\right)\right) \end{cases},$$

where $w\left(s,c_{(q)}\left(a,b\right)\right)=\frac{1-c_{(q)}(a,b)}{1-s}$. Equivalently to the k-optimality in Eq. 3, we **define** a pairwise preference function:

$$F_{\texttt{EL-IS}}\left(a,b\right)=\max_{s}\left(a\preceq_s b\right). \tag{5}$$

Outranking Aftermath. A directed graph with edge weights possessing minimal values of 0 (indicating that a is certainly not better than b) and otherwise larger positive values (indicating the degree to which a is preferred over b) is then constructed based upon one of the three approaches (Eqs. 3, 4 or 5). Note that the outcomes of the three proposed pairwise preference functions are dependent upon their parameter settings: calibrating the *fuzzy membership functions*, or setting the parameters $\{p_j, q_j, v_j\}$ for ELECTRE is critical and is likely to be problem-dependent. While the *fuzzy scoring* concept constitutes a strong tool, the *k-optimality* approach as a whole seems to lack the consideration of the LA perspective, but rather to strongly reflect the GP perspective. Also, the measures $\{n_b, n_e, n_w\}$ do not entirely estimate the DM's utility function because summation over losses versus gains does not capture well one's preferences. On the other hand, the ELECTRE techniques do capture the LA perspective and possess the potential to identify incredibility. At the same time, when a solution-point a is potentially preferred over b, the credibility index becomes inaccurate – especially in cases of a few objectives (up to 4 or 5): minor *loss* variations may change the concordance value (affecting s or σ, respectively).

A Novel Hybrid Perspective. In order to address those effects and yet to benefit from both methods, we propose to hybridize ELECTRE with *fuzzy k-optimality* as follows: utilize ELECTRE to measure incredibility (accounting for LA), and employ *fuzzy* membership functions to evaluate preference (accounting for GP). Here is the proposed pairwise preference function:

$$F_{\text{HYB}}\left(a,b\right)=\begin{cases} 0 & \text{if } \sigma\left(a,b\right)<\theta_{LA} \\ F_{\text{GP}}\left(a,b\right) & \text{otherwise} \end{cases}. \tag{6}$$

We are now left with setting the LA threshold, θ_{LA}, and devise a fuzzy function that estimates the GP preference. To this end, we propose a fuzzy linear function that computes the average gain as an estimator for the DM's utility function: $F_{\text{GP}}\left(a,b\right)=\min\left\{0,\dfrac{n_b-n_w}{m}\right\}$.

3.3 Selection

In the final step of our proposed framework we are to obtain the subset of recommended solutions given the preferences graph. As a reference, in some ELECTRE variants the recommended subset is chosen to be the *graph kernel* $\mathcal{K}$[1], yet following a different graph construction. In our reckoning, kernels do not directly fit the current framework for several reasons. Firstly, they are defined for *unweighted graphs* while our preference graph is *weighted*; secondly, their size is fixed and the proposed recipe wishes to freely set the size of the recommended subset; and finally, not every graph has a kernel (for instance, a directed cycle with $2n - 1$ vertices has an empty kernel). We consider two selection tracks:

1. GP track: solutions that ”win the most” form the *top offensive team*
2. LA track: solutions that ”lose the least” form the *top defensive team*

A naïve line of action would examine the edges around each vertex, and calculate its *degree of optimality*. In the GP track, it refers to the maximal gain over other solution-points, i.e., $deg(v) = \max_{e \in \delta_{out}(v)} w_e$. In the LA track, it refers to minimizing the maximal preference of another solution-point over it, $deg(v) = \max_{e \in \delta_{in}(v)} w_e$. Eventually, solution-points with the highest degree are recommended. The problem with this naïve line of action is that each solution is independently scored, and the obtained subset is not likely to be diverse. We would like to aim for a subset of recommended solutions being good representatives of the Frontier, and we do not wish to solely rely on clustering in achieving diversity. Next, we propose selection lines of action per each track.

GP Track: The Top Offensive Team. Motivated by the graph kernel idea, we attempt at finding a suitable definition to our recommendation case. Intuitively, the definition of recommendation seems to be about selecting a solution subset of a given size n, which optimally ”covers” all solution-points. Thus, we try to relax the notion of a *Dominating Set* for weighted graphs as follows. For each set $\mathcal{D}$, we define the *covering degree* of each vertex as,

$$cvr\left(\mathcal{D}, v\right) = \begin{cases} 1 & \text{if } v \in \mathcal{D} \\ \max\limits_{u \in \mathcal{D}, (u,v) \in E} w(u,v) & \text{otherwise} \end{cases} \cdot \tag{7}$$

We define the *covering degree* of each set as the total degree of all vertices,

$$cvr\left(\mathcal{D}\right) = \sum_{v \in V} cvr\left(\mathcal{D}, v\right). \tag{8}$$

Although the independence property of the kernel is not explicitly expressed here, the choice of maximization, rather than summation, in the definition of $cvr\left(\mathcal{D}, v\right)$ prevents

[1] The kernel $\mathcal{K}$ of a graph is a subset of vertices that is both independent (that is, if $u, v \in \mathcal{K}$, then $(u, v), (v, u) \notin E$), and dominating (that is, for any $v \notin \mathcal{K}$, there exists $u \in \mathcal{K}$ such that $(u, v) \in E$).

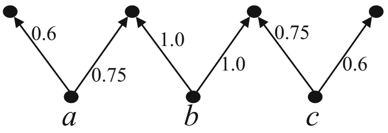


Fig. 3. Illustrating a scenario in the SubDominatingSet selection mode (Eq. 9) where the "most dominating" solution-point is not necessarily picked

us from choosing two similar solutions (the marginal gain of the second solution would be relatively small). An alternative approach that we do not pursue here is to promote diversity by selecting solution-points with "victories" in all coordinates (e.g., by satisfying the $\bigvee_{i=1}^{m}$ operator). We associate the recommended subset with the solution of the following optimization problem:

$$\max_{|\mathcal{D}|\leq n} cvr\left(\mathcal{D}\right) \tag{9}$$

We formalize this problem by means of an integer program. Given the desired number of recommended solutions N_{rec}, let $w(u,v)$ be the (preference) weighted edge between solution-points u and v. We consider binary decision variables $\texttt{sol}[1..|V|]$ to represent the selection of a solution-point (vertex), and binary decision variables $\texttt{cover}[1..|V|][1..|V|]$ where $\texttt{cover}[v][u]=1$ translates to selecting vertex v to cover vertex u. **P1** in Fig. 4, which is a mixed-integer linear program (MILP), realizes Eq. 9. Even though global domination over the entire graph is guaranteed, there is a chance that the "most dominant" solutions will not be selected upon solving **P1**, as illustrated in Fig. 3: Evidently, the best single vertex to be picked therein is b since it possesses a covering degree of 3. At the same time, when targeting a subset of size 2, any solution that contains b has a degree of at most 4.6, while the degree of $\{a,c\}$ is 4.7.

Maximizing the Submodular Coverage Function in a Greedy Approach. An alternative approach would be to generate $\mathcal{D}$ greedily, i.e., to select in each iteration a vertex that maximizes the covering degree considering the solutions that were already chosen. This may degrade the total quality of the subset, but would not miss out the most dominant solutions. At the practical level, the *greedy* approach possesses a straightforward implementation, featuring the following step for each iteration $k=0\ldots N_{\text{rec}}-1$ (setting $\mathcal{D}^0 \leftarrow \emptyset$):

$$\mathcal{D}^{k+1} \leftarrow \mathcal{D}^k \cup \left\{\arg\max_{v\in V}\left\{cvr\left(\mathcal{D}^k \cup \{v\}\right) - cvr\left(\mathcal{D}^k\right)\right\}\right\} \tag{10}$$

Since solving Eq. 9 constitutes maximization of a *submodular monotone function*, **utilizing Eq. 10 guarantees a** $(1-1/e)$**-approximation to it [19]**.

LA Track: The Top Defensive Team. An alternative selection of a recommended subset of solutions would be the defensive angle. Here, we aim at identifying a subset of solutions, ideally diverse, which are not outranked altogether by a single solution. For each *resisting set* $\mathcal{R}$, we define the degree of each vertex as

$$deg\left(\mathcal{R},u\right) = \sum_{v\in\mathcal{R}\setminus u} w\left(u,v\right). \tag{11}$$

[P1] maximize $\sum_{v\in V}\sum_{u\in V} w(u,v)\cdot\texttt{cover}[v][u]$

subject to:

$$\sum_{v\in V}\texttt{sol}[v] \le N_{\text{rec}}$$
$$\sum_{v\in V}\texttt{cover}[v][u] \le 1\ \forall u\in V$$
$$\texttt{cover}[v][u] \le \texttt{sol}[v]\ \forall u\in V\ \forall v\in V$$

[P2] minimize $\texttt{t}$

subject to:

$$\sum_{v\in V}\texttt{sol}[v] \ge N_{\text{rec}}$$
$$\sum_{v\in V} w(u,v)\cdot\texttt{sol}[v] \le \texttt{t}\ \forall u\in V$$

Fig. 4. Mixed-integer linear programs realizing Eqs. 9 and 13

The *resistance degree* of each set is then defined as the maximal degree of all vertices (i.e., the *strongest offense* on $\mathcal{R}$):

$$res\left(\mathcal{R}\right) = \max_{u\in V} deg\left(\mathcal{R}, u\right). \tag{12}$$

In a min-max fashion, we describe the recommended resisting subset as the solution to the following optimization problem:

$$\min_{|\mathcal{R}|\ge n} res\left(\mathcal{R}\right). \tag{13}$$

Given the desired number of solutions N_{rec}, let $w(u,v)$ be the weighted edge between solution-points u and v. We consider binary decision variables $\texttt{sol}[1..|V|]$ to represent a selection of a solution-point, and a dummy continuous decision variable $\texttt{t}$. **P2** in Fig. 4 constitutes a realization to Eq. 13. A *greedy* approach can also be employed in the defensive context. In an analogous way, it is easy to compute the subset of solution-points having the strongest resistance degrees with the following step for each iteration $k = 0 \ldots N_{\text{rec}} - 1$ (setting $\mathcal{R}^0 \leftarrow \emptyset$):

$$\mathcal{R}^{k+1} \longleftarrow \mathcal{R}^k \cup \left\{ \arg\min_{v} \sum_{\mathcal{R}^k\cup\{v\}} w\left(u,v\right) \right\} \tag{14}$$

A Note on Duality. One may think that LA is a dual of GP in a sense that any method to solve one problem can easily be transformed to the other (e.g., one can switch the direction of the edges and get a solution to a GP problem with an LA-solver). This intuition fails as the following example shows – assume a star sub-graph (i.e., there exists a vertex with edges directed at all other vertices), then a GP with $N_{\text{rec}} = 1$ would select it as a singleton and guarantee an optimal solution regardless of the edges between all the other vertices. Upon reversing the edges' direction and solving for LA we would get some subset of the other vertices depending upon the edges amongst them.

4 Summary

We introduced a novel interactive approach for traversing over multiobjective landscapes post-optimization. The proposed traversal is meant to facilitate guided exploration from a reference suboptimal solution to Pareto dominant solutions, which constitute improvements in the objective-space and yet are within baby-steps in the design-space. This proposed navigation, which was derived in both batch and online versions, is targeted at gaining the DM's confidence in the machine-driven Pareto-optimal solutions. At the same time, the inherent tradeoff between the need for adjusting to new conceptual designs to the limited mental resource that characterize DMs was raised. We then proposed a novel MCDM automated recommendation process that has the capacity to facilitate both LA and GP states-of-mind. We derived multiple candidate formulations for describing the outranking relations amongst solution-points on the Efficient Frontier, and devised a recipe for identifying the top subsets by means of combinatorial optimization or greedy approximations. It should be stressed that determining the DM tendency to be GP or LA is beyond the scope of the current work, and is left for future work on the cognitive aspects of such recommender systems.

References

[1] Masin, M., Bukchin, Y.: Diversity maximization approach for multiobjective optimization. Operations Research 56(2), 411–424 (2008)

[2] 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)

[3] Köksalan, M., Wallenius, J., Zionts, S.: Multiple Criteria Decision Making: From Early History to the 21st Century. World Scientific (2011)

[4] Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Trade-offs. John Wiley and Sons, New York (1976)

[5] Saaty, T.L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill (1980)

[6] Roy, B.: The Outranking Approach and the Foundations of ELECTRE Methods. Theory and Decision 31, 49–73 (1991)

[7] Das, I.: A Preference Ordering Among Various Pareto Optimal Alternatives. Structural Optimization 18, 30–35 (1999)

[8] Mattson, C., Mullur, A., Messac, A.: Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space. Engineering Optimization 36(4), 721–740 (2004)

[9] Stolze, M., Ströbel, M.: Dealing with Learning in eCommerce Product Navigation and Decision Support: The Teaching Salesman Problem. In: Proc. of the 2nd Interdiscip. World Congress on Mass Customization and Personalization (2003)

[10] Eskelinen, P., Miettinen, K., Klamroth, K., Hakanen, J.: Pareto Navigator for Interactive Nonlinear Multiobjective Optimization. OR Spectrum 32(1), 211–227 (2010)

[11] Hadzic, T., O'Sullivan, B.: Critique Graphs for Catalogue Navigation. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 115–122. ACM, New York (2008)

[12] Levav, J., Heitmann, M., Herrmann, A., Iyengar, S.S.: Order in Product Customization Decisions: Evidence from Field Experiments. Journal of Political Economy 118(2), 274–299 (2010)

[13] Marinescu, R., Razak, A., Wilson, N.: Multi-Objective Influence Diagrams. In: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 574–583 (2012)

[14] Dijkstra, E.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1(1), 269–271 (1959)

[15] Zadorojniy, A., Masin, M., Greenberg, L., Shir, O.M., Zeidner, L.: Algorithms for Finding Maximum Diversity of Design Variables in Multi-Objective Optimization. Procedia Computer Science 8, 171–176 (2012), Conf. on Sys. Eng. Research

[16] Kahneman, D., Tversky, A.: Prospect Theory: An Analysis of Decision Under Risk. Econometrica 47(2), 263–291 (1979)

[17] Farina, M., Amato, P.: Fuzzy Optimality and Evolutionary Multiobjective Optimization. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 58–72. Springer, Heidelberg (2003)

[18] Lloyd, S.: Least Squares Quantization in PCM. Information Theory 28(2), 129–137 (1982)

[19] Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An Analysis of Approximations for Maximizing Submodular Set Functions - I. Mathematical Programming 14(1), 265–294 (1978)

Cell Mapping Techniques for Exploratory Landscape Analysis

Pascal Kerschke[1], Mike Preuss[1], Carlos Hernández[2], Oliver Schütze[2], Jian-Qiao Sun[3], Christian Grimme[1], Günter Rudolph[4], Bernd Bischl[5], and Heike Trautmann[1]

[1] Department of Information Systems, University of Münster, Germany
{kerschke,christian.grimme,mike.preuss, trautmann}@uni-muenster.de
[2] Department of Computer Science, CINVESTAV, Mexico City, Mexico
chernandez@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx
[3] School of Engineering, University of California, Merced, USA
jsun3@ucmerced.edu
[4] Department of Computer Science, TU Dortmund University, Germany
guenter.rudolph@tu-dortmund.de
[5] Statistics Department, TU Dortmund University, Germany
bischl@statistik.tu-dortmund.de

Abstract. Exploratory Landscape Analysis is an effective and sophisticated approach to characterize the properties of continuous optimization problems. The overall aim is to exploit this knowledge to give recommendations of the individually best suited algorithm for unseen optimization problems. Recent research revealed a high potential of this methodology in this respect based on a set of well-defined, computable features which only requires a quite small sample of function evaluations. In this paper, new features based on the cell mapping concept are introduced and shown to improve the existing feature set in terms of predicting expert-designed high-level properties, such as the degree of multimodality or the global structure, for 2-dimensional single objective optimization problems.

Keywords: exploratory landscape analysis, cell mapping, black-box optimization, continuous optimization, single objective optimization, algorithm selection.

1 Introduction

For the optimization of difficult black-box problems, *Evolutionary Algorithms* (EA) as well as related other metaheuristics are frequently employed. However, different metaheuristics that should in principle be suitable for solving these problems often reveal enormous performance differences or do not solve some problems at all. Thus, the algorithm selection problem must be taken seriously, which means choosing the right algorithm variant and setting its parameters right. This problem has occupied numerous researchers in the last decade, see, e.g., [1] and [2] for an overview.

The *Exploratory Landscape Analysis* (ELA) approach as detailed in section 2 offers an alternative view onto algorithm analysis by focussing on problem analysis. From a small sample of evaluated search points, we calculate a set of features and learn classifiers in order to deduce what the main properties of the problem are. Then, we can make

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_9, © Springer International Publishing Switzerland 2014

an informed guess about which algorithm should be chosen. One of the advantages of this approach is its extensibility: if new feature ideas come up, they can be seamlessly added to the existing ones. If they characterize aspects of the optimization problems well, our prediction of the problem type should improve. The only problem is that an increasing number of features will make classification more difficult. Therefore, any added feature should capture some problem aspects the standard features miss. In this work, we investigate if features obtained from cell mapping techniques (detailed in section 3) fulfill this requirement. We experimentally show in section 4 that this is indeed the case, even if only a relatively small problem sample is employed. Such a sample could be provided by random or *Latin Hypercube Design* (LHD) based initialization of a metaheuristic optimization algorithm (possibly by repeated initialization for restarts, in an amortized fashion), so that applying ELA with cell mapping features does not come at additional cost and should be used when choosing the proper algorithm for a previously unseen problem.

It should be added that while ELA focuses on real-valued optimization problems, there are related concepts for landscape analysis in combinatorial optimization, e.g., Local Optima Networks [3] for detecting the topology of different basins of attraction.

2 Exploratory Landscape Analysis

Exploratory Landscape Analysis (ELA) aims at characterizing optimization problems by means of cheaply computable features based on systematic sampling. The final goal is the construction of a model which allows for an accurate prediction of the best suited algorithm for an arbitrary optimization problem based on the computed features.

During the last years important steps into this direction for single-objective optimization problems have been made. In [4], the benchmarking framework introduced in [5], was applied to the combined experimental results of the benchmarking black-box optimization problem competition (BBOB, [6]). It was investigated whether representative algorithms exhibit similar behavior within the predefined BBOB function grouping. A set of **high-level features** derived by experts were used to characterize the functions, i.e., the degree (none, low, medium, high) of *multimodality*, *global structure*, *separability*, *variable scaling*, *search space homogeneity*, *basin-sizes*, *global to local contrast* and *plateaus*. Of course those are debatable to a certain extent. Using classification techniques based on the high-level features per function instance two clusters of algorithms could be distinguished.

In order to overcome the subjectivity of the previous approach, computable experimental **low-level features** were introduced in [7] which reflect the landscape properties of a problem. The features are grouped into six low-level feature classes, i.e., measures related to the distribution of the objective function values (*y-Distribution*), estimating meta-models such as linear or quadratic regression models on the sampled data (*Meta-Model*) and measures for the convexity and linearity (*Convexity*). Furthermore, local searches are conducted starting at the initial design points (*Local Search*), the relative position of each objective value compared to the median of all values is investigated (*Levelset*), and numerical approximations of the gradient or the Hessian represent the class of curvature features (*Curvature*). Each class contains a set of sub-features which

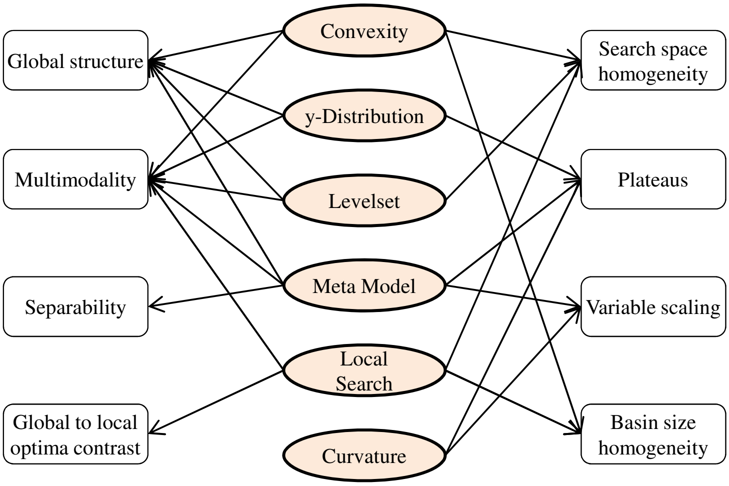


Fig. 1. Relationships between low-level (light orange) standard ELA and high-level (white) features

result from the same experimental data generated from the initial data sample. Fig. 1 visualizes the assumed main relationships between the low-level feature classes and the high-level features introduced in [5].

Experimental validation of these features was conducted based on successfully predicting the values of the high-level from the corresponding low-level features. Both classification accuracy and a cost indicator representing the number of required function evaluations were included into the model building phase. Additionally, the BBOB function grouping could be perfectly predicted by specific combinations of low-level features at moderate cost. Recently, optimal algorithm selection for unseen optimization tasks was addressed in [8] by means of the BBOB09/10 results. A sophisticated cost-sensitive learning approach allowed for accurately predicting the best suited algorithm within a representative algorithm portfolio.

Additional approaches were conducted in [9], based on five features conceptually similar to [7]. In [10], new problem features categorized into the classes *problem definition, hill climbs*, and *random points* were introduced. The concept of *length scale*, which measures the ratio of changes in the objective function value to steps between points in the search space and its distribution, was suggested by [11,12]. A first step into the direction of online algorithm selection based on low-level features is made in [13].

3 Cell Mapping

The cell mapping techniques were originally proposed by Hsu [14]. These methods are useful to determine the global behavior of nonlinear dynamical systems. The main idea of these methods is based on the fact that the representation of the numbers in a computer is finite. According to the precision of the machine, a number does not only represent the number given by its digits, but also an infinite amount of numbers within its neighborhood. The cell mapping approach employs this discretization for dividing the state space into hypercubes. The evolution of the dynamical system is then reduced to a new function, which is not defined in $\mathbb{R}^n$, but rather on the cell space.

Two cell mapping methods have been introduced in order to study the global dynamics of nonlinear systems: *simple cell mapping* (SCM), and *generalized cell mapping* (GCM). The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems [15–17]. Recently, the SCM method has been applied to multi-objective optimization [18]. For more discussions on cell mapping methods, the reader is referred to [14].

3.1 Generalized Cell Mapping

While the SCM offers an effective approach to investigate the global properties of a dynamical system, for problems with complicated characteristics, we need a more sophisticated algorithm. One way is to incorporate more information on dynamics of the system into the cell mapping – which is done in the GCM method. In GCM, a cell z is allowed to have several image cells, being the successors of z. Each of the image cells is assigned a fraction of the total transition probability, which is called the transition probability with respect to z.

The transition probabilities can be grouped into a transition probability matrix P of order $N_c \times N_c$, where N_c is the total number of cells. Then the evolution of the system is completely described by

$$p(n+1) = P \cdot p(n), \tag{1}$$

where p is a probability vector of dimension N_c that represents the probability function of the state. This generalized cell mapping formulation leads to absorbing Markov chains [19].

In the following, we introduce some concepts that are useful to our work.

Absorbing Markov Chain. A Markov chain is absorbing if it has at least one absorbing state, and it is possible to go to an absorbing state from every state (not necessarily in one step).

Classification of Cells. Two types of cells can be distinguished:

A *periodic cell* i is a cell that is visited infinitely often once it has been visited. In our work, we focus on periodic cells of period 1, i.e., $P_{ii} = 1$. This kind of cells correspond to the local optima candidates.

A *transient cell* is by definition a cell that is not periodic. For absorbing Markov chain, the system will leave the transient cells with probability one and will settle on an absorbing (periodic) cell.

Canonical Form (cf). Consider an arbitrary absorbing Markov chain. Renumber the states so that the transient states come first. If there are r absorbing states and t transient states ($N_c = r + t$), the transition matrix has the following canonical form:

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix},$$

where Q is a t by t matrix, R is a nonzero t by r matrix, 0 is an r by t zero matrix, and I is the r by r identity matrix. Matrix Q gathers the probabilities of transitioning from some transient state to another whereas matrix R describes the probability of transitioning from some transient state to some absorbing state.

Fundamental Matrix (fm). For an absorbing Markov chain the matrix $I - Q$ has an inverse $N = (I - Q)^{-1}$. The (i, j)-entry n_{ij} of the matrix N is the expected number of times the chain is in state s_j, given that it starts in state s_i. The initial state is counted if $i = j$. The matrix $fm = I + \sum_{k=1}^{\infty} Q^k$ is called the *fundamental matrix (fm)* of the Markov chain and holds the equation $fm = N$.

Absorbing Probability. This is defined as the probability of being absorbed in the absorbing state j when starting from transient state i, which is the (i, j)-entry of the matrix $B = NR$. In terms of cell mapping, the set of all $B_{i,j} \neq 0$ for $i \in [1, \ldots, t]$ is called the *basin of attraction* of state j, and an absorbing cell within that basin is called *attractor*.

3.2 Adaptation to Exploratory Landscape Analysis

Generalized Cell Mapping. In the following, we assume the problem is bounded by box constraints (lb and ub), which constitutes our domain $Q = \{(x_1, \ldots, x_n)^T \in \mathbb{R}^n : lb_i \leq x_i \leq ub_i,\ i = 1, \ldots, n\}$. Now, we can divide each interval in N_i sections of size $h_i = (lb_i - ub_i)/N_i$. By doing this, we get a finite subdivision of the domain, where each of these elements are called regular cells. The number of regular cells is noted by N_c and we label the set of regular cells with positive integers, $1, 2, \ldots, N_c$. Also, without loss of generality, we will solely consider minimization problems.

One of the drawbacks of GCM is that in the general case, it needs more function evaluations per cell than SCM in order to find the global properties of a dynamical system. However, in the case of optimization, we can use a suitable representative objective value $f(z)$ for each cell z, and then incorporate more information by using the comparison relationship on the set of function values.

Algorithm 1 shows the key elements to compute the characteristics needed to determine the features described within this section. For each cell z, we compare $f(z)$ to the objective values of its neighbors $N_e(z)$. Next, we assign a probability, proportional to their function values, to pass into those cells. If there is no better neighbor cell, equal transition probabilities are assigned to the neighbor cells with equal function values. Worse neighbor cells always get transition probability 0.

A key element of Algorithm 1 is the method of choosing a representative value for $f(s_j)$. In this work, we have chosen the following approaches, based on the available sample points per cell:

- *min_appr:* we select the point with the minimum objective value
- *avg_appr:* we compute the mean of all objective function values
- *near_appr:* we select the function value of the point closest to the center of the cell, even if that point is not in the cell

Please note that in case there are no points in a given cell, only the third approach is computable, whereas the other two would simply fail for these cells.

Algorithm 1. Construction of GCM arguments for single objective optimization

```
Require: $f$: Objective function, $s$: Set of cells
Ensure: $cf$, $fm$
  Compute the set $bc_i = \{s_j | f(s_j) < f(s_i) \text{ for all } s_j \in N_e(s_i)\}$
  Compute the set $pg_i = \{s_j | f(s_j) = f(s_i) \text{ for all } s_j \in N_e(s_i)\}$
  Compute the probability $p_{ij}$ to go from $s_i$ to $s_j$
    $p_{ij} = \begin{cases} (f(s_i) - f(s_j)) \cdot \left( \sum_{k=1}^{|bc_i|} f(s_i) - f(s_k) \right)^{-1} & \text{, if } s_j \in bc_i \\ |pg_i|^{-1} & \text{, if } bc_i = \emptyset \text{ and } s_j \in pg_i \\ 0 & \text{, otherwise} \end{cases}$
  Compute canonical form of $p$, $cf = \begin{bmatrix} I & 0 \\ R & Q \end{bmatrix}$
  Compute fundamental matrix of $cf$, $fm = N = (I - Q)^{-1}$
```

Features. We now present the features that were used in order to characterize the structure of an unknown fitness landscape. In the following, we will call these the *canonical* GCM features. For each of the three approaches (min, average, closest), we consider the following features:

- *Ratio of uncertain cells* (uncert_ratio): is defined as the proportion of cells that lead to different attractors, i.e., the number of non zero entries (nnz) in matrix B at row i which are bigger than 1.

$$uncert\_ratio = \frac{1}{t} \sum_{i=1}^{t} \mathbb{1}_{[\,nnz(B_{i,1:r})>1\,]}$$

- *Probability to find the best cell* (prob_best): the sum of the probabilities of being absorbed by the best cell divided by the total number of cells,

$$prob\_best = \frac{1}{N_c} \sum_{i=1}^{t} B_{i,j},$$

  where j is the absorbing cell with the best function evaluation found.
- *Basin size* (bs): aggregations (standard deviation, minimum, mean and maximum) of the different basin sizes (i.e., the different colored areas in the GCM grids, cf. fig. 2 and 3).
- *Number of attractors* (attr): number of attractors (black boxes) within the grid.

Sometimes, we found considerable differences between the *min_appr* and *avg_appr* approach. In order to study this, we consider the following three features:

- *Common periodic cells* (common.pcells): number of common periodic cells between the approaches. Let $pcell_{avg}$ and $pcell_{min}$ be the periodic cells of the *avg_appr* and *min_appr* approaches respectively, then

$$common.pcells = \left| pcell_{avg} \bigcap pcell_{min} \right| .$$

– *Common transient cells* (common.tcells): number of common transient cells between approaches. Let $tcell_{avg}$ and $tcell_{min}$ be the transient cells of the *avg_appr* and *min_appr* approaches respectively, then

$$common.tcells = \left| tcell_{avg} \bigcap tcell_{min} \right| .$$

– *Percentage of different cells* (common.dcells): the percentage of cells which change their roles (absorbing and transient) from one approach to the other one.

$$common.dcells = 1 - \frac{1}{N_c}(common.pcells + common.tcells)$$

3.3 Examples

In the following, we present two examples using the GCM approach on a 10×10 grid on functions, taken from the BBOB benchmark suite [6]. We employ two approaches for choosing the representative objective function value of each cell: the minimum of each cell (*min_appr*) and the average (*avg_appr*) of the function evaluations.

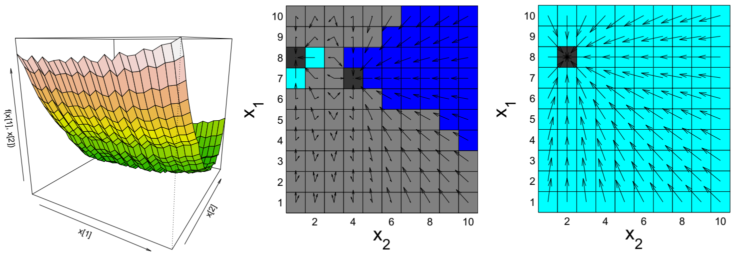
In all figures, dark gray cells represent an attractor, light gray cells are cells that belong to more than one basin of attraction. All other colors refer to a different basin of attraction, and the arrows represent the mappings from one cell to another.

For the Rosenbrock function (fig. 3), both approaches show different characteristics. We can observe that the numbers and locations of attractors are different as well as the sizes of the basins of attraction. In the *min_appr* approach, the parabolic shaped flat valley is reflected by the u-shaped locations of the attractors. For the Rastrigin problem (fig. 2), the *avg_appr* reveals the underlying global structure whereas the *min_appr* provides the main basin of attraction. For the simple, unimodal sphere problem (not shown here), both approaches look like the *avg_appr* of the Rastrigin problem.

3.4 Additional Cell Based Features

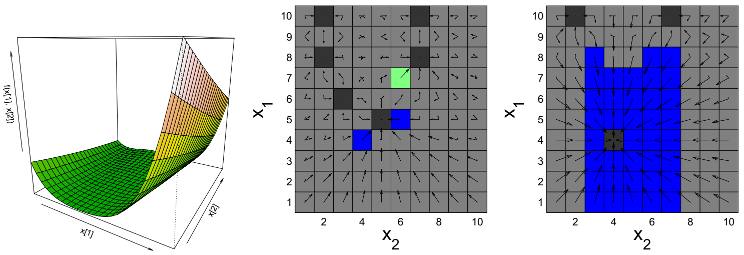
Taking the canonical GCM features in section 3.2 into account, one may look for even more features that use the overall idea of GCM, namely discretization. From [20], we know that the most important high-level ELA features for the partition of the 24 BBOB problems into different groups are *multimodality*, *variable scaling*, and *global structure*. It appears difficult to recognize the *variable scaling* (the deformation of basins due to extreme gradient differences in orthogonal directions) with only a relatively small and evenly distributed sample at hand, because it would be necessary to place multiple points in close vicinity to their neighbors for estimating gradients in different directions.

However, the discretization of a sample into a number of cells, with several points in each cell, opens up possibilities to measure *global structure* and *multimodality*. Note that all features we suggest in this section, except for the last group (*convexity*), are independent of the search space dimensionality. The only precondition to computing them is that we have on average more than one point in each cell. The reason for the independence is that interactions between different cells are ignored, we focus on the cell contents and aggregate the values computed per cell over all cells. The features



(a) graph (b) min_appr (c) avg_appr

Fig. 2. Rastrigin (BBOB-ID 3) with GCM approach



(a) graph (b) min_appr (c) avg_appr

Fig. 3. Rosenbrock (BBOB-ID 8) with GCM approach

defined in section 3.2 may also be transferable to higher dimensions, but this would not be trivial. In the following, we discuss the obtained features in groups that each follow a different concept.

- *Gradient homogeneity* (gradhomo): Fig. 4 visualizes the general idea. For every point within a cell's sample, we find the nearest neighbor and compute the individual, normalized difference vector, which is always rotated so that it is pointing to the worse point. Then, we compute the length of the vector sum of the individual vectors amd divide it by the maximal possible vector length (equals the number of points due to normalization). In the figure, we obtain a value in the range of 0.5, which reflects well that there is a trend for better points towards the bottom of the cell, but there is also some *multimodality*. For completely randomly distributed objective values, the fraction should be around 0 (vectors pointing in all directions), for a strong trend the values should approach 1.0 (all vectors point into the same direction). This is conceptually close to simple step-size adaptation heuristics for the CMA-ES as discussed in [21]. From the individual values for each cell, we

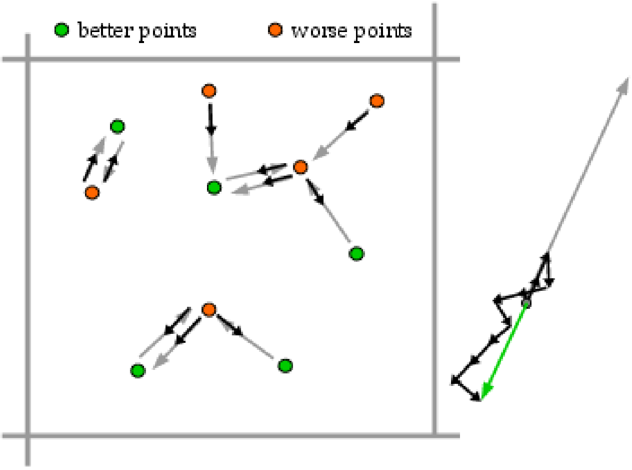


Fig. 4. Schematic view of the *gradient homogeneity* features: for every point, the vector to the nearest point is determined (gray) and normalized so that it points to the better point (black). All black vectors are added (green) and the length of the resulting vector is compared to the added length of all vectors (right).

obtain two features by computing the mean and the standard deviation over all cells. Note that we ignore vector direction differences between cells, only the homogeneity within each cell is taken into account. Simple unimodal functions shall thus generate very high mean values.

- *angle*, *dist_best* and *dist_worst*: Motivated from the previous feature, the location of the best and worst values within the cell might return some insight of the landscape (cf. fig. 5). If they lie in opposite directions it indicates a trend within the cell. In that case the angle between the vectors from cell center to worst value and cell center to best value would be close to 180°. Two features are obtained by aggregating the angles of all cells from the grid using the mean and the standard deviation. Furthermore, the standard deviations in the lengths of the two vectors are used as additional features. In case of simple functions as the *sphere* function, the variation should be low as the majority of the cells have similar distances, because they usually lie close to the borders of the cells. In very multimodal functions, the variation should be high as cells with local optima result in contrary distances (short distances of the best values and long distances of the worst values) compared to cells without any local optima.
- *fun_ratio*: Using the best and worst values provides further information. So far, the features only used the location within the decision space, but their function values were disregarded. Using them, two more features can be obtained. We compute the mean and standard deviation of the distances between the best and worst function values within a cell, scaled by the distance of best and worst function value within the entire grid.

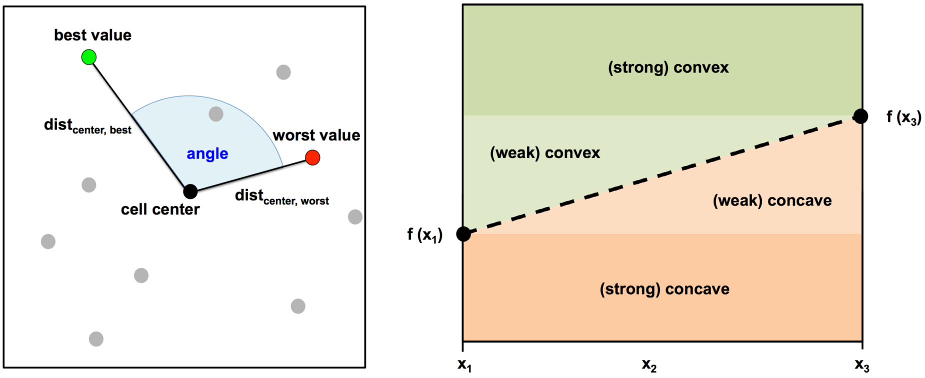


Fig. 5. The length of the vectors from the center to the best and worst value within a cell, as well as the angle between those vectors summarize the direction of the landscape (left). Comparing three (horizontally, vertically or diagonally) neighbouring cells allows to draw conclusions on the local convexity (right).

- *convex_weak*, *convex_strong*, *concave_weak* and *concave_strong*: These four features focus on the convexity of the functions' landscape. For any three (in a line) neighboring cells, the observations, which are located closest to the cell centers (x_1, x_2 and x_3), should also be more or less in a line. A function is said to be (weak) convex, if $f(\alpha \cdot x_1 + (1 - \alpha) \cdot x_3) > \alpha \cdot f(x_1) + (1 - \alpha) \cdot f(x_3)$ for $\alpha \in (0, 1)$. Assuming that x_2 lies approximately in the middle of x_1 and x_3, i.e., $\alpha = 0.5$, the function is convex if $f(x_2) > 0.5 \cdot (f(x_1) + f(x_3))$. Furthermore, it is strong convex, if $f(x_2) > \max\{f(x_1), f(x_3)\}$. The concavity can be derived analogously (cf. fig. 5). Based on that approach, each of the four features can be derived by computing its ratio over all possible combinations of three (either horizontally, vertically or diagonally) neighboring cells within the grid.

4 Experimental Analysis

Our overall goal is to identify the features that enable predicting the high-level properties of an unknown problem in order to select a matching optimization algorithm. In the ideal case, we would aim at a diverse set of decision space dimensions so that we obtain a universally valid classifier. However, the canonical GCM features of section 3.2 can only be computed for 2D problems without a sophisticated redesign of cell location and neighborhoods. We therefore restrict this first analysis to 2D. Although we assume that this setting should be easier than 5D or 10D as attempted on the 24 BBOB functions in [7], we currently have no comparison data available as in that work, the easier case of leave-one-instance-out cross-validation was considered. Thus, our comparison will be a relative one between the standard ELA features, the whole set of new features of section 3.2 and 3.4, and all of these combined. We employ feature forward selection to find well-performing, but small feature sets for each feature group. We aim at detecting for which high-level features (as *multimodality*) the new features actually provide a considerable advantage. The experiments were run using MATLAB [22] (canonical GCM

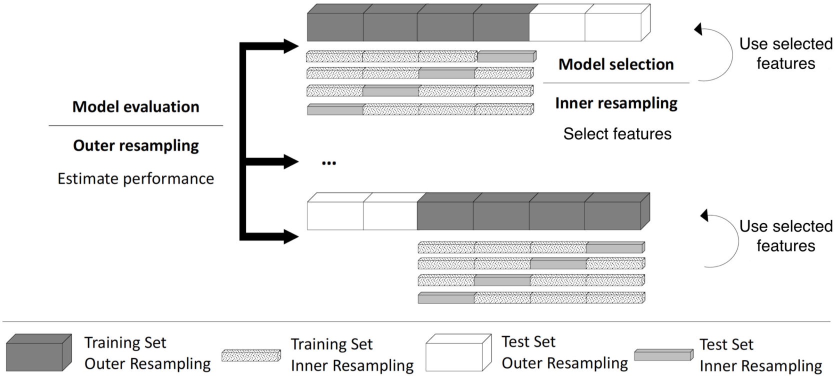


Fig. 6. Nested Resampling strategy for feature selection, inspired by [25]

features) and R [23] (additional GCM features), for resampling and feature selection the R package mlr [24] was employed.

4.1 Experimental Setup

A sample of 1000 evaluations, randomly distributed over $10 \times 10 = 100$ cells in 2D was employed for all considered problems. This is a relatively small number, but still larger than the number of initial samples used, e.g., in most EAs. However, for difficult multimodal problems, 1000 is small in comparison to the number of function evaluations necessary to solve them. Additionally, the surplus from the setup of a start population could be used for the initialization of restarts so that not too many evaluations are wasted.

The experiments are based on the 24 BBOB functions, for each one we select 10 function instances and perform 10 statistical replications. The features were averaged over the replications, providing a reduction of the variance among the stochastic features. Thus, the setup consists of a total of 240 instances. The 50 low-level ELA features, introduced in [5] were reduced to 22, as the feature groups belonging to local search, convexity and curvature were discarded due to their need for additional function evaluations. In addition to those 22 ELA features, 44 GCM features were used: two common cells features, three approaches covering ten features each (cf. section 3.2), and the additional cell based features (cf. section 3.4). Each high-level property will be predicted by a *random forest* classifier (using default settings, i.e., 500 trees [26]). Missing values among any of those 66 features were imputed with a value twice as high as the highest non-missing value within that feature, which is a reasonable and standard imputation technique for tree-based methods.

The model validation [25] is done using a nested resampling strategy as visualized in fig. 6, which is a standard evaluation approach in machine learning for feature selection scenarios as ours. In order to generate a realistic scenario, the functions were blocked for the modeling, i.e., all instances that belong to the same BBOB function were used either for training or testing. This way, the data can be split up into a maximum of 24

Table 1. Classification of the BBOB functions based on their properties (*multimodality*, *global structure*, *separability*, *variable scaling*, *homogeneity*, *basin sizes*, *global-to-local*). Predefined groups are separated by horizontal lines and changes to previous versions are colored red.

Function	multim.	gl.-struc.	separ.	scaling	homog.	basins	gl.-loc.
1: Sphere	none	none	high	none	high	none	none
2: Ellipsoidal separable	none	none	high	high	high	none	none
3: Rastrigin separable	high	strong	high	low	high	low	yes
4: Büche-Rastrigin	high	strong	high	low	high	med.	yes
5: Linear Slope	none	none	high	none	high	none	none
6: Attractive Sector	none	none	none	low	med.	none	none
7: Step Ellipsoidal	none	none	none	low	high	none	none
8: Rosenbrock	low	none	none	none	med.	low	yes
9: Rosenbrock rotated	low	none	none	none	med.	low	yes
10: Ellipsoidal high conditioned	none	none	none	high	high	none	none
11: Discus	none	none	none	high	high	none	none
12: Bent Cigar	none	none	none	high	high	none	none
13: Sharp Ridge	none	none	none	low	med.	none	none
14: Different Powers	none	none	none	low	med.	none	none
15: Rastrigin multimodal	high	strong	none	low	high	low	yes
16: Weierstrass	high	med.	none	med.	high	med.	yes
17: Schaffer F7	high	med.	none	low	med.	med.	yes
18: Schaffer F7 moderately ill-cond.	high	med.	none	high	med.	med.	yes
19: Griewank-Rosenbrock	high	strong	none	none	high	low	yes
20: Schwefel	low	none	none	none	high	low	yes
21: Gallagher 101 Peaks	low	none	none	med.	high	med.	yes
22: Gallagher 21 Peaks	low	none	none	med.	high	med.	yes
23: Katsuura	high	none	none	none	high	low	yes
24: Lunacek bi-Rastrigin	high	none	none	low	high	low	yes

blocks – one per function. Both, the inner and outer loops, use a *leave-one-function-out* (LOFO) *cross-validation* (CV). Thus, the outer loop partitions the data into 24 blocks, each one consisting of one BBOB function (10 instances, colored white in fig. 6) in the test data and the remaining 23 functions (230 instances, dark gray) in the corresponding model selection set. On each of the model selection sets, forward selection is used for selecting the best feature sets. To evaluate a potential feature set, the random forest performance on this feature set is calculated using a LOFO CV (in the inner loop) on the 230 instances of the model selection set. When the feature forward selection has terminated, a random forest is finally trained on the whole model selection set with the selected feature set and its *misclassification error* (MCE) is measured on the corresponding test data. Thus, the resampling strategy returns 24 unbiased performance values and feature sets (one per fold of the outer loop).

It is important to understand that blocking the functions and using the nested resampling approach leads to a more realistic estimation of the MCE as this approach avoids overfitting to the training data. The approach as a whole is different from the one used in [7] and should lead to a less optimistic, but more realistic classification quality assessment.

In order to handle very small classes, which lead to problems during (blocked) cross-validation, some classes within the properties *multimodality* (low and me-dium), *global structure* (deceptive, weak and none) and *global-to-local* (low, medium and high) were

Table 2. Comparison of the MCEs, aggregated using the median and mean over the 24 folds of the outer LOFO CV (best performances written in bold type). Based on a Wilcoxon signed-rank test, the differences between all features and the ELA features were significant for the properties *global structure* and *multimodality* (w.r.t. significance niveau 10%). Also, there were significant differences between ELA and GCM for *multimodality* and *variable scaling*.

property	**Median**			**Mean**		
	all	**ELA**	**GCM**	**all**	**ELA**	**GCM**
Basin Size	0.40	**0.20**	0.35	0.47	**0.31**	0.47
Global to Local	**0.00**	**0.00**	**0.00**	**0.14**	0.16	0.19
Global Structure	**0.00**	0.10	**0.00**	**0.18**	0.34	0.21
Homogeneity	**0.00**	0.35	0.20	**0.28**	0.39	0.34
Multimodality	**0.00**	0.35	**0.00**	**0.15**	0.36	0.20
Separability	**0.00**	**0.00**	**0.00**	**0.17**	0.20	0.24
Variable Scaling	0.25	**0.00**	0.60	0.39	**0.28**	0.53

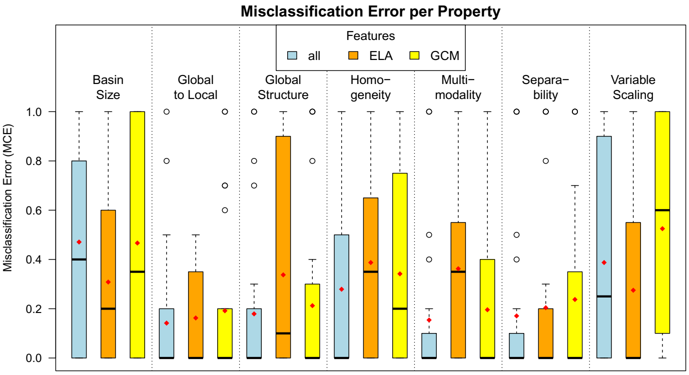


Fig. 7. Boxplots of MCEs per property and feature subset. Each boxplot is based on 24 performance values, obtained during the model evaluation. The red diamonds indicate the mean of each sample, i.e., the MMCE.

merged. The property *plateau* was removed completely as it was a 2-class problem, with one class consisting of only one observation. All used properties are shown in table 1.

4.2 Results and Discussion

Comparing the three performance values (one per feature group, i.e., ELA, GCM and their combination) for each of the seven high-level properties shows, that the GCM features improve the ELA features in five of the seven categories. Table 2 reveals that especially the properties *global structure*, *homogeneity* and *multimodality* benefit from the addition of the GCM features as the corresponding *mean misclassification errors* (MMCEs), i.e., the mean over the MCEs of the 24 folds, were reduced by 10-20%. As the BBOB set was created in a way that each function of that set covers different

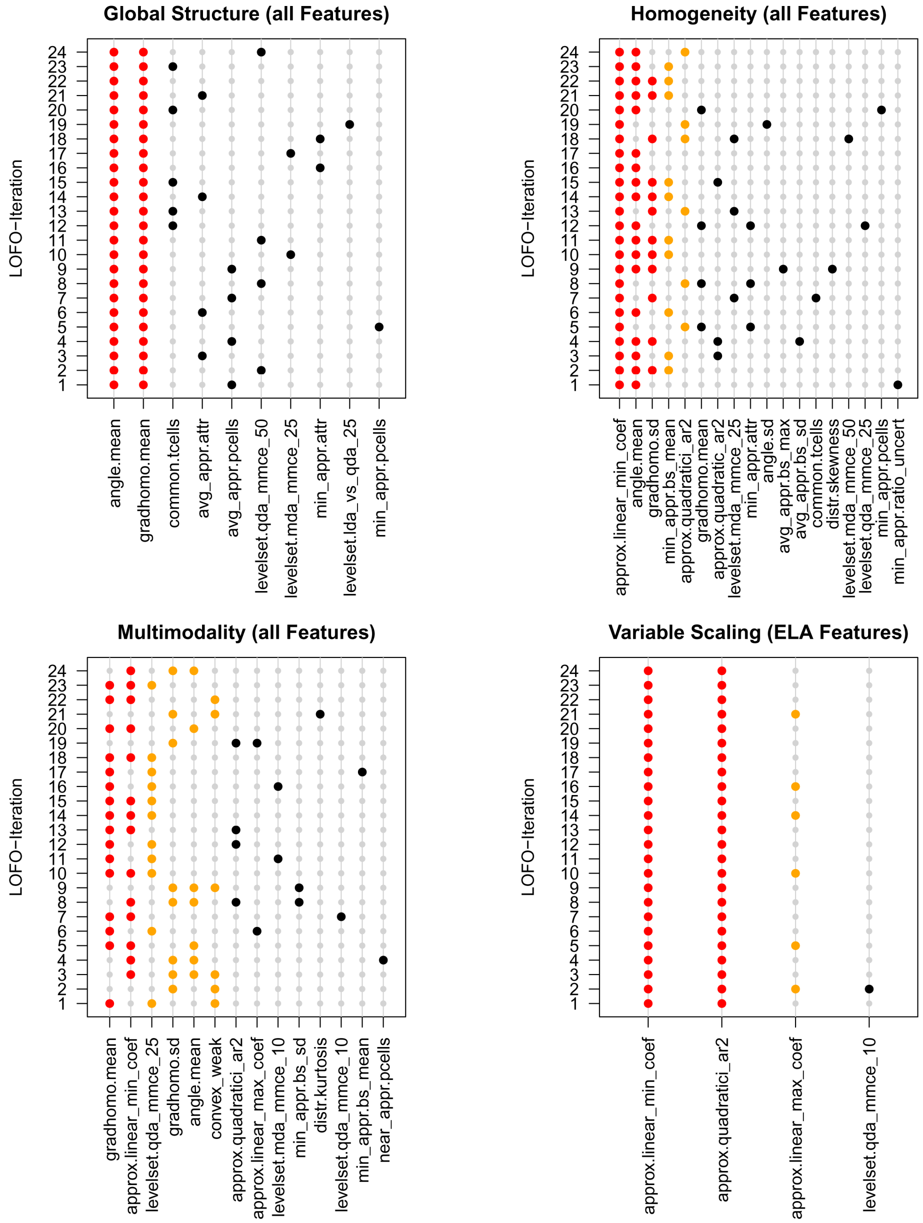


Fig. 8. The figures above show the selected features per fold within the model validation. All the illustrated cases reveal major differences between the feature groups. The color indicates whether the feature was chosen at least once (black), in at least 25% (orange), or in at least 50% (red) of the folds.

aspects, the variance within the misclassification rates is quite high. However, using a Wilcoxon ranked-sum test, it could be shown that the improvements in *global structure* and *multimodality* are statistically significant w.r.t. a significance niveau of 10%, which is remarkable considering the few performance values. Given that none of the GCM features explicitly aims at explaining properties like *variable scaling*, it is very reasonable that the new features were not able to improve the performance of this property. Instead, the performance decreased significantly, probably due to adding redundant features.

It is also noteworthy that the MMCE of four properties is below 20%, which is good w.r.t. the fact that a very strict and realistic validation method (nested resampling with leave-one-function-out cross-validations) was applied.

As mentioned before, each BBOB problem describes a different problem and thus, their characteristics are very diverse. Hence, it is reasonable to compare all performances, e.g., using boxplots (cf. fig. 7), instead of comparing aggregated measurements such as median or mean. Comparing the performances over all 24 folds also reveals that the MCEs are skewed positively, i.e., in the majority of iterations the models are very good and therefore fail only in a few iterations.

Furthermore, one might be interested in the selected features. Due to the nested resampling strategy, 24 (different) feature sets exist, which cannot be aggregated. Instead, it is more reasonable to look at the importance of the selected features, e.g., by analyzing how often each feature was selected. Fig. 8 shows the importance plot for the four cases in which ELA differed strongly from the other feature sets. In matters of *global structure*, the *angle* and *gradient homogeneity* features were selected in each of the 24 subsets and therefore they seem to be the features which mainly describe this property. Also, both of these features are, combined with two *meta model* features (ELA) and another GCM feature, important for explaining the *homogeneity*. Adding those two features towards some *meta model* and *levelset* features also leads to a major improvement in describing the *multimodality* of a function. However, in case of *variable scaling* the ELA features, especially three features from the *meta model* group, provide already sufficient information, which deteriorated by the perturbation of the significantly worse GCM features.

5 Conclusions and Outlook

We have approached the extension of the standard ELA feature set from the perspective of discretization, namely by using *general cell mapping* (GCM) properties as features in order to better predict the high-level properties as *multimodality* and *homogeneity*. Furthermore, we have extended the canonical GCM features by a set of newly designed features that use only the assignment of observations (search points) to cells and are therefore (with exception of the *convexity* features) completely independent of cell location and neighborhood, and thus of the number of dimensions. Whereas it would be nontrivial to extend the canonical GCM features to more than 2D, this requires no change other than the redefinition of cells for the additional features of section 3.4.

For the aforementioned reasons, the experimental analysis focused on 2D with a relatively small sample of 1000 points. The results show that the new features are especially valuable for predicting the high-level properties *multimodality* and *global structure*, which are, according to the original ELA experiments, most important for selecting a proper algorithm for a difficult black-box problem. Especially the new (additional) *angle* and *gradient homogeneity* features are chosen regularly by the feature selection, whereas the canonical GCM features play only a minor role.

This is not only a very good improvement but also reveals how other successful features should be created. As the ELA approach can easily integrate new features, there are endless possibilities for designing features in order to improve the classification even for the 2 (of 7) high-level properties that have not been improved. Additionally, the feature selection process itself shall be investigated and improved further (if feature selection would be perfect, i.e., if finding the best subset of features would be guaranteed, adding more features could never result in deterioration). Simple forward selection is obviously not ideal, but total enumeration is also not possible due to the combinatorial explosion. One shall try more clever heuristics or meta-heuristics such as EAs for further improvement.

Acknowledgements. This work was supported by the grant ANR-12-MONU-0009-03 (NumBBO) of the French National Research Agency. The third author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN.

References

[1] Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.): Parameter Setting in Evolutionary Algorithms. SCI, vol. 54. Springer, Heidelberg (2007)

[2] Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M.: Experimental methods for the analysis of optimization algorithms. Springer (2010)

[3] Ochoa, G., Tomassini, M., Vérel, S., Darabos, C.: A study of nk landscapes' basins and local optima networks. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 555–562. ACM, New York (2008)

[4] Mersmann, O., Preuss, M., Trautmann, H., Bischl, B., Weihs, C.: Analyzing the BBOB Results by Means of Benchmarking Concepts. Evolutionary Computation Journal (accepted for publication, 2014)

[5] Mersmann, O., Trautmann, H., Naujoks, B., Weihs, C.: Benchmarking evolutionary multiobjective optimization algorithms. In: Ishibuchi, H., et al. (eds.) IEEE Congress on Evolutionary Computation (CEC), pp. 1311–1318. IEEE Press, Piscataway (2010)

[6] Hansen, N., Auger, A., Finck, S., Ros, R.: Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA (2010)

[7] Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011, pp. 829–836. ACM, New York (2011)

[8] Bischl, B., Mersmann, O., Trautmann, H., Preuss, M.: Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO 2012, pp. 313–320. ACM, New York (2012)

[9] Muñoz, M.A., Kirley, M., Halgamuge, S.K.: A meta-learning prediction model of algorithm performance for continuous optimization problems. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 226–235. Springer, Heidelberg (2012)

[10] Abell, T., Malitsky, Y., Tierney, K.: Features for exploiting black-box optimization problem structure. In: Nicosia, G., Pardalos, P. (eds.) LION 7. LNCS, vol. 7997, pp. 30–36. Springer, Heidelberg (2013)

[11] Morgan, R., Gallagher, M.: Length scale for characterising continuous optimization problems. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 407–416. Springer, Heidelberg (2012)

[12] Morgan, R., Gallagher, M.: Using landscape topology to compare continuous metaheuristics: A framework and case study on edas and ridge structure. Evol. Comput. 20(2), 277–299 (2012)

[13] Muñoz, M.A., Kirley, M., Halgamuge, S.K.: Landscape characterization of numerical optimization problems using biased scattered data. In: IEEE Congress on Evolutionary Computation, pp. 1–8. IEEE (2012)

[14] Hsu, C.S.: Cell-to-cell mapping: A method of global analysis for nonlinear systems. Applied mathematical sciences. Springer (1987)

[15] 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)

[16] 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)

[17] 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)

[18] Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., Sun, J.-Q.: Simple Cell Mapping Method for Multi-objective Optimal Feedback Control Design. International Journal of Dynamics and Control 1(3), 231–238 (2013)

[19] Kemeny, J., Snell, J.: Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix". Undergraduate Texts in Mathematics. Springer (1976)

[20] Mersmann, O., Preuss, M., Trautmann, H.: Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 73–82. Springer, Heidelberg (2010)

[21] Jägersküpper, J., Preuß, M.: Empirical investigation of simplified step-size control in metaheuristics with a view to theory. In: McGeoch, C.C. (ed.) WEA 2008. LNCS, vol. 5038, pp. 263–274. Springer, Heidelberg (2008)

[22] MATLAB: version 8.2.0.701 (R2013b). The MathWorks Inc., Natick, Massachusetts (2013)

[23] R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013)

[24] Bischl, B.: mlr: Machine Learning in R. R package version 1.2

[25] Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods in model validation. Evolutionary Computation Journal 20(2), 249–275 (2012)

[26] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer (2009)

Parallel Cell Mapping for Unconstrained Multi-Objective Optimization Problems

Jesús Fernández Cruz[1], Oliver Schütze[1], Jian-Qiao Sun[2], and Fu-Rui Xiong[3]

[1] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, C.P. 07360,
Col. San Pedro Zacatenco, Mexico City, México
jfernandez@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx

[2] University of California Merced
School of Engineering
Merced, CA 95344, USA
jsun3@ucmerced.edu

[3] Department of Mechanics
Tianjin University
Tianjin, 300072, China
xfr9311@gmail.com

Abstract. Recently, the cell mapping techniques, originally designed for the global analysis of dynamical systems, have been proposed as a numerical tool to thoroughly investigate multi-objective optimization problems. These methods, however, suffer the drawback that they are restricted to low-dimensional problems, say $n \leq 5$ decision variables. The reason is that algorithms of this kind operate on a certain discretization on the entire search space resulting in a cost that is exponential to n.

As a possible remedy we propose in this paper to use a parallel implementation of the cell mapping strategy. We will make use of Graphics Processing Units (GPUs) which allow for comfortable speedups in particular for problems that are costly to evaluate since cell mapping techniques are highly parallelizable. We will test our methods on several widely used benchmark models with $n = 10$ decision variables and make some comparisons to the sequential version of the cell mapping technique.

Keywords: multi-objective optimization, cell mapping techniques, parallel computing, Graphics Processing Unit (GPU).

1 Introduction

In multi-objective optimization, the task is to simultaneously optimize several objectives that arise in a given application. As a general example, we mention product design, where two typical tasks are to maximize the quality of the given product while minimizing its overall cost.

In literature, a huge variety of methods dealing with the numerical treatment of MOPs can be found. There are, for instance, many scalarization methods which transform the MOP into a 'classical' scalar optimization problem (SOP). By choosing a clever sequence of SOPs a suitable finite size approximation of the entire Pareto set can

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_10, © Springer International Publishing Switzerland 2014

be obtained (see [1–4] and references therein). Since the Pareto set forms under some mild regularity conditions locally a $(k-1)$-manifold, specialized continuation methods which perform a search along the Pareto set are very efficient if one (or more) solution is at hand ([5–8]). Another approach to approximate the Pareto set is to use set oriented methods such as subdivision techniques ([9–11]) or stochastic search methods such as evolutionary algorithms (e.g., [12, 13] and references therein).

The cell mapping techniques, which we will consider in this study, fall into the latter category. Cell mapping techniques were originally designed for the global analysis of general dynamical systems ([14, 15]). Algorithms of that type discretize the entire search space into cells and consider instead of the point-wise iteration of the dynamical system f the dynamics on the cells induced by f. In [16, 17] cell mapping techniques have been adapted to the numerical treatment of MOPs.

Though the obtained results were very promising one severe drawback of them is that they are restricted to the treatment of low-dimensional problems due to the discretization of the entire search space.

Parallel computing is a paradigm of computation in which many calculations are processed simultaneously. The interest in parallelism has grown due to the physical constraints of frequency scaling in order to obtain performance gains. Instead, parallel computing uses multiple processing elements simultaneously to solve a given problem. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others.

Graphics Processing Units (GPUs) are high-performance many-core processors capable of very high computation and data throughput ([18]). They were initially designed for computer graphics and could only be programmed through relatively complex application programming interfaces (APIs) like DirectX ([19]) and OpenGL ([20]). Nowadays, GPUs are general-purpose processors with specially designed APIs like CUDA ([21]) and OpenCL ([22]). Applications may obtain great speedups even when compared against finely tuned central processing unit (CPU) implementations. GPUs are currently used in a wide array of applications, including gaming, bioinformatics, chemistry, finance, numerical analysis, imaging, weather, etc. Such applications are usually accelerated significantly, and accelerations of 10x or more are common ([23]). Parallel computer programs are in general more difficult to write than sequential ones, since concurrency introduces several new classes of potential errors, of which race conditions are the most common ones. Communication and synchronization between the different subtasks are typically some of the greatest obstacles in getting good performance.

A program can be parallelized in two ways: *Data parallelism* where the data is distributed across different parallel computing nodes; or *task parallelism* where the computer code is distributed across multiple processors.

Nowadays, parallel computing is widely used for the numerical treatment of MOPs in many works due to the high execution time. Some examples are surveyed in the following.

In [24] the authors propose the use of GPU-based parallel computing as a complementary way to speed up the search. They also present a new methodology to design and implement efficiently and effectively hybrid evolutionary algorithms on GPU accelerators. The methodology enables efficient mappings of the explored search space onto the GPU memory hierarchy. The experimental results show that the approach is very efficient especially for large problem instances.

Also in [25] the authors propose to use GPUs to increase the speedup of the original algorithms.

Research in this area as e.g. led to a text book ([26]) to give light of this advanced computational technique, in particular addressing generic local search, tabu search, genetic algorithms, differential evolution, swarm optimization, ant colony optimization, systolic genetic search, genetic programming, and multi-objective optimization.

In this paper, we argue that the use of GPUs is advantageous for the parallelization of the cell mapping techniques. The core of the method, namely to compute the mapping from one cell to the next one, is highly parallelizable, in particular, since this operation is rather simple to evaluate in the GPU in most cases: Starting from the center point of each box B, the underlying dynamical system is followed for a certain stretch, and the box that contains the end point of the evaluation is the designated follow-up box of B in the cell dynamics. Then an analysis of the dynamics must be performed in order to get the candidate cells. Finally, a dominance test is computed to remove all the dominated cell of the archive, which yields the obtained Pareto set.

Here we show that cell mapping techniques can be used for the treatment of MOPs with dimension $n = 10$ with great accuracy and within reasonable time (e.g., ranging between one and two seconds for the benchmark models considered in this study).

The remainder of this paper is organized as follows: In Section 2, we briefly state the required background for the understanding of this paper. In Section 3, we present the GPU based cell mapping techniques for the treatment of MOPs. In Section 4, we present some numerical results on some commonly used benchmark functions for $n = 10$ decision variables and report the obtained speedups compared to the sequential cell mapping. Finally, we conclude in Section 5 and give some possible paths for future research.

2 Background

In the following we consider MOPs of the form

$$\min_{x \in Q} \{F(x)\}, \tag{MOP}$$

where $Q \subset \mathbb{R}^n$ is the domain and F is defined as the vector of the objective functions $F : \mathbb{R}^n \to \mathbb{R}^k$, $F(x) = (f_1(x), \ldots, f_k(x))$, and where each objective $f_i : \mathbb{R}^n \to \mathbb{R}$ is (for simplicity) sufficiently smooth. Though we in principle do not handle constrained problems here, we have for the applicability of the cell mapping techniques to assume that Q is given by an n-dimensional box, i.e.,

$$Q = \{x \in \mathbb{R}^n \; : \; l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n\}, \tag{1}$$

where $l \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ are the lower and upper bounds, respectively.

The optimality of a MOP is defined by the concept of *dominance* ([27]): A vector $y \in \mathbb{R}^n$ is *dominated* by a vector $x \in \mathbb{R}^n$ ($x \prec y$) with respect to (MOP) if $f_i(x) \leq f_i(y)$, $i = 1, \ldots, k$, and there exists an index j such that $f_j(x) < f_j(y)$, else y is non-dominated by x. A point $x \in \mathbb{R}^n$ is called *(Pareto) optimal* or a *Pareto point* if there is no $y \in \mathbb{R}^n$ which dominates x. The set of all Pareto optimal solutions is called the *Pareto set*, and is denoted by $\mathcal{P}$. The image $F(\mathcal{P})$ of the Pareto set is called the *Pareto front*. Both sets typically form a $(k-1)$-dimensional object.

In order to measure the perfomance of a parallel program we can use Amdahl's law ([28]) which indicates the maximum possible speedup of a program as a result of parallelization.

Definition 1. *The maximum acceleration which can be obtained given n processors is represented by:*

$$S(n) = \frac{1}{p_s + \frac{1}{n}(1 - p_s)}, \tag{2}$$

where $p_s \in [0, 1]$ is the sequential part of the program, i.e. the part that cannot be parallelized.

Amdhal's law also states that there exists an upper bound for the accelaration regardless of the number of processing elements available.

Definition 2. *The maximum theoretical acceleration (upper bound) is given in the limit where the number of processors approaches infinity:*

$$\lim_{n \to +\infty} S(n) = \lim_{n \to +\infty} \frac{1}{p_s + \frac{1}{n}(1 - p_s)} = \frac{1}{p_s}, \tag{3}$$

where $p_s \in [0, 1]$ is the sequential part of the program, i.e. the part that cannot be parallelized.

Gustafson's law ([28]) says that computations involving arbitrarily large data sets can be efficiently parallelized. Gustafson's law provides a counterpoint to Amdahl's law fixed data set size.

Definition 3. *According to Gustafson's law, the maximum acceleration which can be obtained given n processors is represented by:*

$$S(n) = n - p_s(n - 1), \tag{4}$$

where $p_s \in [0, 1]$ is the sequential part of the program, i.e. the part that cannot be parallelized.

Gustafson's law addresses the shortcomings of Amdahl's law, which does not fully exploit the computing power that becomes available as the number of machines increases. Gustafson's law instead proposes that programmers tend to set the size of problems to use the available equipment to solve problems within a practical fixed time. Therefore, if faster (more parallel) equipment is available, larger problems can be solved in the same time.

3 The Algorithms

Here we describe a possible parallel implementation of the cell mapping techniques for the treatment of MOPs.

The cell mapping techniques as used in [16, 17] have some nice properties with respect to their parallelization:

- The evolution (or mapping) of each cell is completely independent from the evolution of the other cells.
- The parallelizable part of the program is larger than its non-parallelizable (sequential) part, and this difference increases as the cell number grows.

The two main tasks in cell mapping which are also the most costly ones are (i) the creation part and (ii) the evolution part. Further, there exist additional tasks such as the analysis of p-groups, the dominance test, among others. Due to its data dependance between the creation and the evaluation part, we cannot expect great accelerations if we use task parallelism. However, this dependance is removed if we use data parallelism which we suggest here.

In the following we describe the elements of the parallel cell mapping procedure and will further on present the complete algorithm.

The first step is to create the set of cells together with their center points to be processed by the algorithm (see Algorithm 1 for the pseudo code of *Creation*). The key factor of this algorithm is the function $Getid$ which must return a unique and valid value, which is the identity id of a cell c_{id}. Each cell has a center point cp and a function value fv.

Algorithm 1. Algorithm Creation

```
Require: Function $F$, total cells $tot$ in the system, sizes $h$ per dimension, and limits $lim$ per dimension.
$id \leftarrow Getid(0)$
$j \leftarrow 0$
while $id \leq tot$ do
    $c_{id}.cp \leftarrow$ Center point of the cell.
    $c_{id}.fv \leftarrow f(c_{id}.cp)$
    $j \leftarrow j + 1$
    $id \leftarrow Getid(j)$
end while
```

Once the algorithm created on GPU memory all the required information we can proceed with the evaluation of each cell (see Algorithm 2). As for Algorithm *Creation*, we need to calculate a unique and valid id for the cell. Then, we calculate the mapping defined by the underlying dynamical system for the cell c_{id}. Based on this, we decide if the cell is mapped into another cell $\tilde{c}$ (i.e., a new vector is found dominating the center point of the current box which is located in cell $\tilde{c}$) or the cell is mapped to itself.

It is important to remember that the division of the problem is over the data set (each cell is processed individually on each core), thus the acceleration is given by the numbers of cells m_c in the system

$$S(n) = \frac{1}{\frac{1}{m_c} + \frac{1}{n}(1 - \frac{1}{m_c})} \tag{5}$$

Algorithm 2. Algorithm Evaluation

```
Require: Number $tot$ of total cells in the system, the sizes $h$ per dimension, and their limits $lim$ per dimension.
$id \leftarrow Getid(0)$
$j \leftarrow 0$
while $id \leq tot$ do
    Calculate the decent direction $v$
    $t \leftarrow norm(h)$
    $aux \leftarrow f(c_{id}.cp + t * v)$
    while $c_{id}.fv \prec aux$ do
        $t \leftarrow t/2$
        $aux \leftarrow f(c_{id}.cp + t * v)$
    end while
    $c_{id}.nextCell \leftarrow i$ s.t. $aux \in c_i$
    $j \leftarrow j + 1$
    $id \leftarrow Getid(j)$
end while
```

In this way, we can proceed with the analysis algorithm (see Algorithm 3) in order to mark the candidate cells where the non-dominated set might be located in. Algorithm *Analysis* goes through all cells and checks if they belong to the p-group. In principle, only 1-groups are of interest (i.e., self-mapping cells). Note, however, that by discretization of the cell mapping ansatz also p-groups with $p > 1$ can be generated that are of interest (the most common ones are 2-groups that perform a ‘flipping’ around a (local) solution). The algorithm marks all p-groups as candidates when their period is less or equal to a maximum value mp.

Finally, a parallel dominance test has to be performed using the center points of all candidate cells, see Algorithm 4. Cells those center points are not dominated by any other center points of the candidate set are stored as part of the (obtained) Pareto set. The Pareto front can then e.g. be approximated via considering the function values of the center points.

The above algorithms constitute the required steps to perform the parallel cell mapping for a fixed discretization of the parameter space. Note, however, that even if we achive a great performance due to the GPUs, we still face the same dimensionality problem as for the sequential algorithm, namely that the number of cells (and thus the overall cost of the algorithm) increases exponentially with the number of dimensions. which will eventually overflow the memory capabilities of any computer. Here, we follow the suggestion made in [29] and perform the cell mapping in an iterative way on

Algorithm 3. Algorithm Analysis

```
Require: Total cells $tot$ in the system and the maximal period $mp$.
$id \leftarrow Getid(0)$
$j \leftarrow 0$
while $id \leq tot$ do
    $i \leftarrow 0$
    $c = i \leftarrow c_{id}.nextCell$
    while $c_{id} \neq c_i$ and $i < mp$ do
        $c = i \leftarrow c_{id}.nextCell$
    end while
    if $c_{id} = c_i$ then
        mark $c_{id}$.
    end if
    $j \leftarrow j + 1$
    $id \leftarrow Getid(j)$
end while
```

Algorithm 4. Algorithm Dominance Test

```
Require: Total candidate cells $tot$ and a set $C$ of candidates cells.
$id \leftarrow Getid(0)$
$j \leftarrow 0$
while $id \leq tot$ do
    for all $c \in C$ do
        if $c_{id} \prec c$ then
            Mark the cell.
        end if
    end for
    if $c_{id}$ was not marked then
        Save $c_{id}$
    end if
    $j \leftarrow j + 1$
    $id \leftarrow Getid(j)$
end while
```

the set of candidate cells. In this way, cell mapping gets combined with subdivision techniques presented in [9–11].

Algorithm 5 presents the overall iterative parallel version of Simple Cell Mapping (SCM) for the treatment of multi-objective optimization problems. The main idea is to make a gross SCM trying to use all the available GPU processors. Then, the best points selected by the dominance test constitute the new search space. This process, cell mapping and selection, is iterated until the desired depth md of the search is reached.

Algorithm 5. Algorithm pSCM

```
Require: Total cells $tot$ in the system, the sizes $h$ per dimension and their limits $lim$, the depth $d$ of the search, and the maximal depth $md$.
Ensure: Approximation of the Pareto set and Pareto front of the MOP.
$cs \leftarrow \emptyset$
if $d = md$ then
    Store candidate set $cs$.
else
    Create($f$,$tot$,$h$,$lim$)
    Evaluate($f$,$tot$,$h$,$lim$)
    SystemAnalysis($tot$,2)
    $cs \leftarrow$ DominanceTest($tot$)
    for all $c_i \in cs$ do
        $h \leftarrow h/tot$
        pSCM($f$,$tot$,$h$,$c_i.lim$,$d+1$,$md$)
    end for
end if
```

4 Numerical Results

Here we present some numerical results of the novel parallel cell mapping algorithm and discuss the speedup against its sequential counterpart. For the problems, we have chosen to take the bi-objective ones from the ZDT ([30]) and DTLZ ([31]) benchmark suites. These problems have different characteristics (e.g., convex, concave or disconnected Pareto front, uni- or multi-modal functions, Pareto front within or at the boundary of the domain) and are widely used to demonstrate the capability of an algorithm to compute the entire Pareto front of a given MOP. In all cases we have used $n = 10$ for the dimension of the parameter space. For the computations, we have used an Alienware Mx17 R4 laptop with 8GB RAM and the following characteristics: (i) a Nvidia gtx 680m GPU with 1344 CUDA cores and a 720 MHz processor clock, (ii) an Intel Core i7-3720QM CPU with a clock speed of 2.6 GHz and a maximal turbo frequency of 3.6 GHz. The design parameters of pSCM for each problem can be found in Table 2.

Figure 1 shows the numerical approximations of the Pareto fronts of the ZDT functions plus the true Pareto fronts and Figure 2 the respective ones for the DTLZ functions. Apparently, in all cases the approximations are nearly identical to the true Pareto sets and can certainly be considered to be ‘good enough’ from the practical point of view. This observation is underlined by Table 2 which shows the obtained Δ_p values (for $p = 2$) of the obtained approximations and the true Pareto fronts. The indicator Δ_p ([32]) can be viewed as an averaged Hausdorff distance between two sets. Since all values are close to zero, this indicates a good approximation quality of the outcome sets.

In the following we investigate the acceleration obtained via pSCM compared to its sequential counterpart. Table 2 shows the execution times for the creation and the evaluation algorithm as well as for the complete procedure. The total acceleration for these seven examples ranges from a speedup of four to nearly seven. This is on the one hand not that overwhelming. Due to the overhead associated with transitions between GPU

Table 1. Number of restarts and divisions per coordinate direction for the MOPs considered in this study

	restarts	divisions per coordinate
$ZDT1$	12	[2 2 2 2 2 2 2 2 2 2]
$ZDT2$	12	[2 2 2 2 2 2 2 2 2 2]
$ZDT3$	6	[5 3 3 3 3 3 3 3 3 3]
$ZDT4$	4	[5 5 5 5 5 5 5 5 5 5]
$ZDT6$	12	[2 2 2 2 2 2 2 2 2 2]
$DTLZ1$	12	[2 2 2 2 2 2 2 2 2 2]
$DTLZ2$	6	[5 3 3 3 3 3 3 3 3 3]

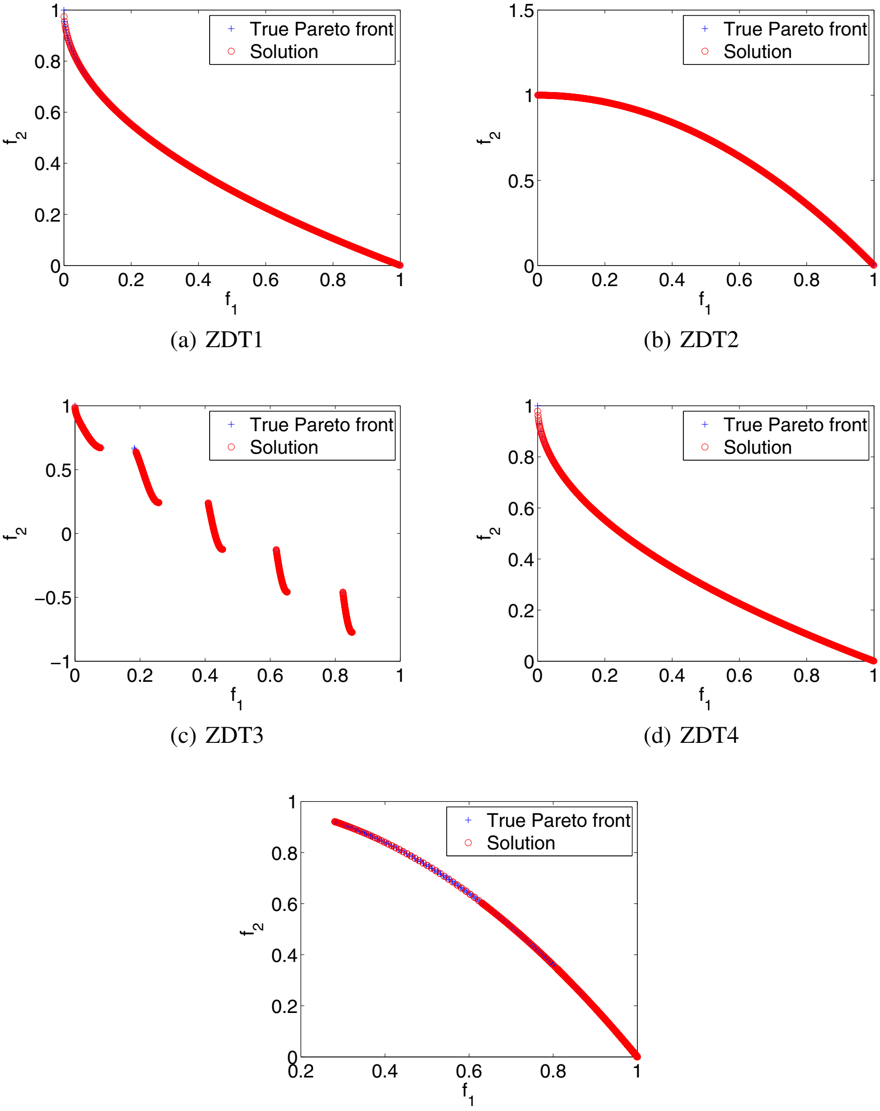
Table 2. Obtained Δ_2 values of the approximations obtained by pSCM and the true Pareto fronts

	Δ_p value
$ZDT1$	0.0043
$ZDT2$	0.0043
$ZDT3$	0.0035
$ZDT4$	0.0043
$ZDT6$	0.0037
$DTLZ1$	0.0011
$DTLZ2$	0.0046

Table 3. Computational times and speedups obtained by SCM and pSCM on the benchmark models. All the times are in milliseconds.

	Creation		Evaluation		Total		Total acceleration
	parallel	sequential	parallel	sequential	parallel	sequential	
$ZDT1$	150.78	1367.52	642.99	3861.02	1345.11	5461.8	4.06x
$ZDT2$	157.94	2025.54	646.44	4318.52	1270.69	6989.44	5.50x
$ZDT3$	144.18	1705.36	642.99	3861.02	1326.13	6244.94	4.70x
$ZDT4$	274.72	3235.67	677.76	6727.01	1506.64	10216.8	6.78x
$ZDT6$	216.87	2740.5	851.26	7047.48	1688	10101.29	5.98x

and CPU, the practical acceleration of each problem is (much) below the theoretical one. On the other hand, we note that parallel computing allows for computational times of 1.3 to 1.7 seconds which is more than reasonable for the global solution of problems of that dimension. Further, it has to be noted that the evaluation time for each of the chosen academic models is quite low (less than a millisecond per function evaluation). For real-world applications, for which the cell mapping techniques are originally designed, such evaluation times are not realistic, but can range in minutes or even hours as e.g. for airfoil design ([33, 34]). In order to artificially increase the cost of our benchmark functions without defining new ones, we evaluate each function m times in each cell.



(a) ZDT1

(b) ZDT2

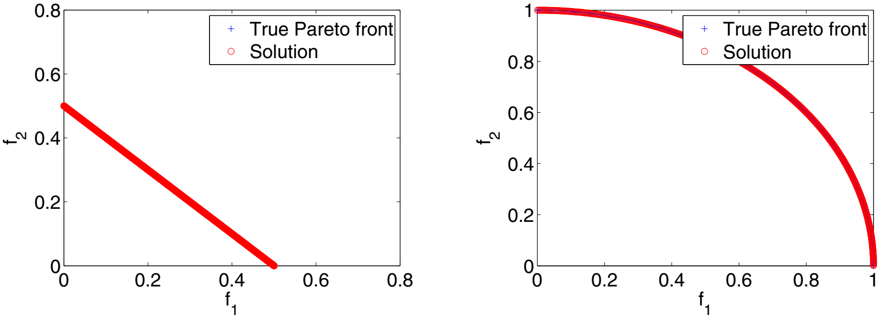
(c) ZDT3

(d) ZDT4

(e) ZDT6

Fig. 1. Numerical results of pSCM on the five continuous models of the ZDT benchmark suite

Tables 3 to 5 show the total acceleration obtained for the values $m = 10, 100$ and 1000 (even for $m = 1000$ the evaluation time for each function is still much less than one second). The main reason for the difference in the accelerations shown in the tables is due to logarithmic and trigonometric operations. Inside the GPU, these operations are performed in hardware units called *Special Functions Units* which are fewer than the



(a) DTLZ1

(b) DTLZ2

Fig. 2. Numerical results of pSCM on the DTLZ benchmark models

Table 4. Computational times and speedups for $m = 10$ function evaluations to increase the cost of the evaluation. All the times are in milliseconds.

	Creation		Evaluation		Total		Total acceleration
	parallel	sequential	parallel	sequential	parallel	sequential	
$ZDT1$	216.78	4100.29	885.33	9544.66	1653.45	13878.21	8.39x
$ZDT2$	908.58	8305.58	947.844	14753.33	2353.334	23268.37	9.88x
$ZDT3$	855.21	5561.39	870.34	12059.91	2309.36	17837.43	7.72x
$ZDT4$	1785.53	23592.76	1508.6	36939.37	3841.77	60789.04	15.82x
$ZDT6$	1271.08	19319.85	1593.42	40400.23	3535.89	60045.2	16.98x

Table 5. Computational times and speedups for $m = 100$ function evaluations to increase the cost of the evaluation. All the times are in milliseconds.

	Creation		Evaluation		Total		Total acceleration
	parallel	sequential	parallel	sequential	parallel	sequential	
$ZDT1$	773.86	28397.96	2377.93	57299.13	3690.73	85962.74	23.29x
$ZDT2$	7336.26	64829.31	4033.77	104667.53	11888.89	169704.77	14.27x
$ZDT3$	6873.71	39873.82	2882.07	77183	10274.94	117319.35	11.41x
$ZDT4$	15543.47	206302.48	9310.72	306838.34	25409.73	513415.04	20.20x
$ZDT6$	10311.19	161705.26	8578.83	335403.28	19511.11	497455.11	25.49x

number of cores in the GPU. However, as can be seen speedups of more than 30 can be obtained. For instance, the computational time for ZDT6 is about 3 minutes when using pSCM, while the sequential SCM needs 81 minutes for the same computation. Thus, one can say that pSCM can obtain reasonable time savings and speedups if more realistic models are under consideration which is scope of future studies.

Table 6. Computational times and speedups for $m = 1000$ function evaluations to increase the cost of the evaluation. All the times are in milliseconds.

	Creation		Evaluation		Total		Total acceleration
	parallel	sequential	parallel	sequential	parallel	sequential	
$ZDT1$	6031.4	270900.7	17609.6	532799.7	24198.26	803979.4	33.22x
$ZDT2$	71108.5	631137.9	34965.96	1001487.8	106581.5	1632867.0	15.32x
$ZDT3$	67032.0	383867.53	23360.0	729958.5	90903.2	1114083.9	12.25x
$ZDT4$	152470.2	2039935.3	87354.85	3056862.0	240392.0	5097091.6	21.20x
$ZDT6$	100382.3	1588335.8	78485.2	3282948.5	179502.0	4871649.8	27.13x

5 Conclusions and Future Work

In this paper, we have proposed pSCM, a parallel implementation of the cell mapping technique for the approximation of the Pareto set/front of a given multi-objective optimization problem. Due to the data dependency of its two most costly tasks, creation and evolution, we have chosen for a data parallelism. The implementation has been designed for the use of GPUs which allows for the use of a large number of cores on a relatively small and cheap computer. We have presented numerical results on seven state-of-the-art benchmark models with $n = 10$ decision variables. The results show that reasonable time savings could have been obtained compared to its sequential counterpart and that further reasonable speedups can be expected in case the function evaluations get more costly.

For the future, our main focus will be to apply pSCM for the treatment of real-world problems such as airfoil design ([33,34]) where rather high computational times for the evaluation of a system are expected. This will call among others for the transportation of the computation to a more advanced architecture. Further, it is expected that the constraint handling techniques have to be improved such as the additional consideration of equality constraints. Finally, we intend to apply the parallel algorithm to other problems such as constrained multi-objective optimization, and bi-level optimization where the cell mapping ansatz will be advantageous.

Acknowledgement. The first author acknowledges support from CONACyT through a scholarship to pursue undergraduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 128554.

References

[1] Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization 8, 631–657 (1998)

[2] Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)

[3] Fliege, J.: Gap-free computation of Pareto-points by quadratic scalarizations. Mathematical Methods of Operations Research 59, 69–89 (2004)

[4] Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Heidelberg (2008) ISBN 978-3-540-79157-7

[5] Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer (1990)

[6] Hillermeier, C.: Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach. Birkhäuser (2001)

[7] Pereyra, V.: Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems. Math. Comput. Simul. 79(6), 1935–1947 (2009)

[8] Wang, H.: Zigzag search for continuous multiobjective optimization. INFORMS J. on Computing 25(4), 654–665 (2013)

[9] Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. Journal of Optimization Theory and Applications 124, 113–155 (2005)

[10] Jahn, J.: Multiobjective search algorithm with subdivision technique. Computational Optimization and Applications 35(2), 161–175 (2006)

[11] Schütze, O., Vasile, M., Junge, O., Dellnitz, M., Izzo, D.: Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. Engineering Optimization 41(2), 155–181 (2009)

[12] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X

[13] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007) ISBN 978-0-387-33254-3

[14] Hsu, C.: Cell-to-cell mapping: a method of global analysis for nonlinear systems. Applied mathematical sciences. Springer (1987)

[15] 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)

[16] Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., Sun, J.Q.: Simple cell mapping method for multi-objective optimal feedback control design. International Journal of Dynamics and Control 1(3), 231–238 (2013)

[17] Hernández, C., Sun, J.-Q., Schütze, O.: Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 171–188. Springer, Heidelberg (2013)

[18] General-Purpose Computation on Graphics Hardware (October 2013), http://gpgpu.org/

[19] Getting Started with DirectX Graphics (January 2014), http://msdn.microsoft.com/en-us/library/hh309467

[20] Open GL wiki (January 2014), www.opengl.org/wiki/

[21] NVIDIA Documentation (October 2013), http://docs.nvidia.com/cuda/index.html

[22] Open CL - The open standard for parallel programming of heterogeneous systems (January 2014), https://www.khronos.org/opencl/

[23] GPU Applications (October 2013), http://www.nvidia.com/object/gpu-applications-domain.html

[24] Van Luong, T., Melab, N., Talbi, E.: Parallel hybrid evolutionary algorithms on GPU. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)

[25] Fok, K.L., Wong, T.T., Wong, M.L.: Evolutionary computing on consumer-level graphics hardware. IEEE Intelligent Systems 2, 69–78 (2007)

[26] Tsutsui, S., Collet, P.: Massively Parallel Evolutionary Computation on GPGPUs. Natural Computing Series. Springer (2013)

[27] Pareto, V.: Manual of Political Economy. The MacMillan Press (1971 (original edition in French in 1927))

[28] Rauber, T., Rünger, G.: ParallelProgramming For Multicore and Cluster Systems. Springer-Lehrbuch. Springer (2007)

[29] Naranjani, Y., Hernández, C., Xiong, F.-R., Schütze, O., Sun, J.-Q.: A hybrid algorithm for the simple cell mapping method in multi-objective optimization. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 207–223. Springer, Heidelberg (2013)

[30] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)

[31] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, pp. 105–145. Springer, USA (2005)

[32] 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 16(4), 504–522 (2012)

[33] Kipouros, T.: Stochastic optimisation in computational engineering design. In: Schütze, O., Coello Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., Legrand, P. (eds.) EVOLVE - A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation II. AISC, vol. 175, pp. 475–490. Springer, Heidelberg (2012)

[34] Oliver, J.M., Kipouros, T., Savill, A.M.: A self-adaptive genetic algorithm applied to multi-objective optimization of an airfoil. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 261–276. Springer, Heidelberg (2013)

Part V

Local Search and Optimization

Adaptive Multi-operator MetaHeuristics for Quadratic Assignment Problems

Madalina M. Drugan[1] and El-Ghazali Talbi[2]

[1] Artificial Intelligence lab, Vrije Universiteit Brussels, Pleinlaan 2, 1050-Brussels, Belgium
[2] Univ. Lille, CNRS and INRIA Lille Nord Europe Parc Scientifique de la Haute Borne 40,
Avenue Halley Bt.A, Park Plaza 59650 Villeneuve d'Ascq, France

Abstract. Local search based algorithms are a general and computational efficient metaheuristic. Restarting strategies are used in order to not be stuck in a local optimum. Iterated local search restarts the local search using perturbator operators, and the variable neighbourhood search alternates local search with various neighbourhoods. These two popular restarting techniques, or operators, evolve independently and are disconnected. We propose a metaheuristic framework, we call it *multi-operator metaheuristics*, which allows the alternative or simultaneously usage of the two restarting methods.

Tuning the parameters, i.e. the neighbourhood size and the perturbation rate, is essential for the performance of metaheuristics. We automatically adapt the parameters for the two restarting operators using variants of *adaptive pursuit* for the multi-operators metaheuristic algorithms.

We experimentally study the performance of several instances of the new class of metaheuristics on the quadratic assignment problem (QAP) instances, a well-known and difficult combinatorial optimization problem.

1 Introduction

Metaheuristics. [1,2] is a general, successful and powerful search method for difficult optimization problems. Local search (LS) based metaheuristics starts from an initial solution and iteratively generates new solutions using a neighbourhood strategy. Each step, a solution that improves over the existing best-so-far solution is chosen. The local search stops when there is no possible improvement, i.e. in a local optimum. Because LS can be stuck in local optima, some advanced local search algorithms consist in restarting the LS.

We consider two basic techniques to restart the local search. *Variable neighbourhood search* (VNS) [3,4] is a variant of local search that changes the neighbourhood function to escape local optimum. A set of neighbourhood functions are alternated either in a predefined or uniform randomly order.

Multi-restart local search (MLS) restarts local search multiple times from uniform randomly chosen initial solutions in order to find different basin of attractions in different part of the search space. There are certain limitations in the design of multi-restart LS because it is basically random sampling in the space of local optima, it does not scale up for large number of local optima.

To improve upon multi-restart LS's performance, stochastic local search aims to escape from local optimal sets by stochastic perturbation operators that preserve partial

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_11, © Springer International Publishing Switzerland 2014

information of the perturbed solutions. In case of *iterated LS* (ILS) [2], the perturbation operator is mutation. It has been pointed out [5, 6] that the experimental performance is sensitive to the choice of the mutation' parameters.

The two different techniques restart the local search using two different parameters: i) the neighbourhood functions for VNS, and ii) the mutation rate for the iterated LS. Section 2 gives some background on metaheuristic algorithms and we show a detailed example of metaheuristics for the quadratic assignment problem (QAP).

Quadratic Assignment Problem (QAP). Intuitively, QAPs can be described as the optimal assignment of a number of facilities to a number of locations. QAPs are NP-hard combinatorial optimization problems that model many real-world problems (i.e., scheduling, vehicle routing). Let us consider N facilities, a set $\Pi(N)$ of all permutations of $\{1, \ldots, N\}$ and the $N \times N$ distance matrix $A = (a_{ij})$, where a_{ij} is the distance between location i and location j. We assume a flow matrix $B = (b_{ij})$ where b_{ij} represents the flow from facility i to facility j. The goal is to minimize the *cost function*

$$c(\pi) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \cdot b_{\pi(i)\pi(j)} \tag{1}$$

where $\pi(\cdot)$ is a permutation. It takes quadratic time to evaluate this function.

The Main Contributions. In general, certain metaheuristic algorithm instances are preferred to solve specific problem instances. For example, iterated local search is commonly used to optimize QAP instances, but seldom used with variable size neighbourhood. There is no study to indicate when and for which instances one of the two methods is superior and, even more, if the alternative or simultaneously use of the two restarting methods is beneficial or not.

We propose a class of metaheuristic that allows the use of both operators to restart the search: i) variable neighbourhood search, and ii) stochastic perturbator operators. We call this the *Multi-operator MetaHeuristics* (MMH) problem and we introduce it in Section 3.

We assume that these two restarting techniques are complementary but with the same goal of finding better local optimum. The stochastic perturbator operators exploit the structure of the search space, and thus they can be considered the *diversification* technique of LS. The variable neighbourhood search is the *intensification* technique for the MMH problem because it explores the search space by changing the size of the neighbourhood.

The multi-operator metaheuristic problem has several parameters to tune. The parameters of the variable neighbourhood search are the size of the neighbourhoods, and we need a strategy to decide when and how to alternate the neighbours. The iterated local search has similar parameters that are the size of the stochastic perturbator operator(s). Again, we need a strategy to decide which mutation rate to use. In addition, an adaptive multi-operator metaheuristic algorithm needs a strategy to decide when and how to use each restarting method.

Tuning the parameters, and in particular tuning parameters of the restarting strategies for metaheuristics, is a complex process that is preferable to carry out automatically. *Pursuit allocation strategies* (AP) [7] are on-line operator selection algorithms that adapt a selection probability distribution such that the operator with the maximal estimated reward is often pursued. The second contribution of this paper is the generalization of the adaptive pursuit algorithm to adapt more than one parameter, e.g. the size of the neighbourhood and the mutation step. We call this *multi-operator adaptive pursuit* and introduce it in Section 4. We assume that the two parameters are correlated and their alternatively or simultaneously usage helps the exploration of the search space. Although we apply this type of algorithms on multi-restart adaptive metaheuristics, the scope of the multi-operator adaptive pursuit is broader. For example, it could be used to simultaneously adapt the mutation and recombination operators of evolutionary algorithms, or any other set of operators of an evolutionary algorithm.

In Section 5, we show preliminary experimental results of the proposed algorithms on several instances of the *quadratic assignment problem* (QAP) [8,9]. Section 6 concludes the paper.

2 Metaheuristics: Background

In this section, we present background knowledge on two, up to date, independent metaheuristic algorithms. Iterated Local search (ILS) [10] is a very popular local search based metaheuristic because of its simplicity and the ease in usage. Variable Neighbourhood search (VNS) [3] is one of the first metaheuristic popular in solving specific type of problems like the min-max combinatorial optimization problems. We show an effective implementation of local search for QAP instances, which is a permutation problem.

The common part of the two metaheuristics is the local search function. Consider that local search as a combination of: i) a neighbourhood function, and ii) a neighbourhood exploration strategy, or an improvement strategy.

The Neighbourhood Function, $\mathcal{N}$, generates solutions in the neighbourhood of a given solution s. $\mathcal{N}(s)$ has as input a solution s and returns the set of neighbours for that solution. Thus, $\mathcal{N} : \mathcal{S} \leftarrow \mathcal{P}(\mathcal{S})$ is a function that maps the solution space $\mathcal{S}$ to sets of solutions $\mathcal{P}(\mathcal{S})$. The neighbourhood function depends on the problem (e.g., quadratic assignment problem) is applied on.

A suitable neighbourhood function for QAPs is the 2-exchange swapping operator that swaps two different facilities. This operator is attractive because of its linear time to compute the difference in the cost function with the condition that the flow and distance matrices are symmetrical.

The size of a neighbourhood increases quadratically with the number of facilities. This means that for a 2-exchange swapping operator, there are $\binom{N}{2}$ neighbours for a solution, and for a m-exchange swapping operator, there are $\binom{N}{m}$ for a single solution. We call this the m-th *neighbourhood exchange rate*.

The Neighbourhood Exploration Strategy, $\mathcal{I}$, decides when to stop the expansion of a neighbourhood $\mathcal{N}(s)$. We consider $\mathcal{I}(s, \mathcal{N})$ a tuple of solutions and neighbourhood functions. There are mainly two neighbourhood exploration techniques [11]: i) the first improvement, and ii) the best improvement. The *best improvement* explores *all* the individuals in the neighbourhood of a solution and selects the best solution that improves

Algorithm 1. Iterated local search (ILS)

```
$t \leftarrow 0$
$s' \leftarrow$ GenerateInitialSolution
$s^{(0)} \leftarrow$ LocalSearch($s'$)
while the stopping criteria is NOT met do
    $s' \leftarrow$ Perturbation($s^{(t)}$)
    $s'' \leftarrow$ LocalSearch($s'$)
    if $s''$ is an improvement over $s^{(t)}$ then
        $s^{(t+1)} \leftarrow s''$
    end if
    $t \leftarrow t + 1$
end while
return The local optimum solution $s^{(t)}$
```

over the initial solution of a neighbourhood. The *first improvement* stops when the first improvement to the initial solution is found.

Note that the best improvement strategy does not depend on the problem because the entire neighbourhood is expanded. The first improvement strategy depends on the definition of the improvement [6].

For the QAP problems, which is a minimization problem, a solution s is considered better than another solution s' iff $c(s) < c(s')$. The cost function to optimize is $c(\cdot)$, where $c : \mathcal{S} \leftarrow \mathbb{R}$, and $\mathcal{S}$ is the solution space and $\mathbb{R}$ is the real valued space.

Furthermore, for large neighbourhoods, the first improvement strategies are more efficient than the best improvement strategies which will spend considerable amount of times in the neighbourhood of the first, suboptimal solutions. Also experimentally, the first improvement based metaheuristics are acknowledged to outperform the best improvement metaheuristics for some combinatorial optimization problems [6, 11].

2.1 Iterated Local Search (ILS)

Iterator local search [10] restarts local search from solutions generated with perturbator operators, like the mutation operator, in order to escape the local optimum. The neighbourhood function, $\mathcal{N}$, is assumed to be fixed, e.g. for the QAP instances is a 2-exchange operator. The pseudo-code for iterated local search is presented in Algorithm 1.

The ILS algorithm starts with a uniform randomly generated solution $s^{(0)}$. A local search function is called on this randomly generated solution, LocalSearch($s^{(0)}$). With a standard ILS algorithm, this is the only local search call started from a uniform randomly generated solution. Until a stopping criteria is met, we perform the following loop which has a counter t. A new solution s' is generated from the current local optimum solution $s^{(t)}$ using a perturbator operator. We assume that the generated solution is a sub-optimal solution. The local search is restarted from the new solution and the returned solution s'' replace the current local optimum $s^{(t)}$ iff it is better than $s^{(t)}$. The algorithm returns the best found so far local optimum.

Algorithm 2. Local search $\mathsf{LS}(s, (\mathcal{N}_1, \ldots, \mathcal{N}_P))$

```
$t \leftarrow 0$
while An improvement is possible do
    $\mathcal{N}^{(t)} \leftarrow \mathsf{SelectNeighbourhood}(\mathcal{N}_1, \ldots, \mathcal{N}_P)$
    $s' \leftarrow \mathcal{I}(s^{(t)}, \mathcal{N}^{(t)})$
    if $s'$ is an improvement over $s^{(t)}$ then
        $s^{(t)} \leftarrow s'$
    end if
    $t \leftarrow t + 1$
end while
return The local optimum solution $s^{(t)}$
```

The Perturbator Operator. Like the neighbourhood function, the perturbator operator depends on the problem instance. In permutation problems like QAPs, the *mutation operator* exchanges facilities between different positions. The m-exchange mutation uniform randomly selects m distinct pairs of positions which are sequentially exchanged.

We assume a mutation exchange rate that is larger than the neighbourhood exchange rate. When LS uses the 2-exchange operator to generate a neighbourhood, the mutation operator should exchange *at least* 3 facilities to escape from the region of attraction of a local optimum.

In practice, this mutation operator is often tuned. A small m could mean that the local search cannot escape the basin of attraction. A large m means basically random sampling in the search space and then the structure of the search space cannot be exploited. The optimal value for m depends on the landscape of the search space [5], i.e. the size of the basin of attractions.

The stopping criterion of the iterated LS is chosen to fairly compare its performance with the multi-restart LS. The search in the iterated LS is halted when it reaches the same number of swaps as the multi-restart LS. The distance between two solutions is defined as the minimum number of exchanges necessary to obtain one solution from another. The distance between a solution and its m-exchange solution is $m-1$. Counting the number of swaps is equivalent with counting the number of function evaluations.

The Acceptance Criteria. The solution returned by local search s'' is accepted, iff it improves the current local optimum solution, $s^{(t)}$. Thus, if a solution is accepted, then $s^{(t)} \leftarrow s''$. Recall that for the QAP instances, a solution s'' is better than $s^{(t)}$ iff $c(s'') < c(s^{(t)})$.

2.2 Variable Neighbourhood Search (VNS)

VNS [4] is a metaheuristic that systematically change its neighbourhood function in order to escape from the current local optimal solution. A VNS algorithm considers that two neighbourhood functions started from the same solution can have different local optimal solutions. The global optimal solution is considered the best local optimal solution of all neighbourhood functions.

Local Search. The definition of local search for VNS generalizes the previous definition of local search from Section 2.1 because it uses a set of neighbourhood functions that are alternated during a local search run. The pseudo-code for the local search algorithm is given in Algorithm 2.

The input for the local search requires an initial solution s and a set of P neighbourhood functions $\mathcal{N} = (\mathcal{N}_1, \ldots, \mathcal{N}_P)$. For the local search function from Section 2.1, we have that the set of neighbourhood functions contains only one neighbourhood functions. We use a counter t for the number of neighbourhood exploration functions. Until an improvement is still possible, we select a neighbourhood function, $\mathcal{N}^{(t)} \in (\mathcal{N}_1, \ldots, \mathcal{N}_P)$, from the set of neighbourhoods in order to explore the corresponding neighbourhood of the current solution $s^{(t)}$. If the resulting solution s' is an improvement over $s^{(t)}$, the current solution is replaced by s' and the counter t is updated.

SelectNeighbourhood. There are various techniques to alternate the neighbourhood functions [4]. The most popular VNS variant is the *variable neighbourhood descend* that deterministically changes the size of the neighbourhood and performs a local search with all neighbourhood functions. The resulting local optimum is the optimum for all the neighbourhood functions. The *reduced VNS* changes uniform randomly the neighbourhood functions each time the exploration neighbourhood function is called. The *skewed VNS* variant resemble, in some extend, the multi-restart LS because it explores regions that are far from the initial solution by generating random restarting solutions.

3 Multi-operator Metaheuristics

In this section, we propose a class of metaheuristic algorithms that use two (possible more) schemes that restart the local search. This algorithm is a combination of two metaheuristics, i.e. the iterated local search and the variable neighbourhood search. ILS and VNS have in common the local search function call, but have different restarting strategies. The two restarting strategies can be used simultaneously or alternatively. If the two strategies are used alternatively, another strategy to decide the restarting technique at a time step is needed.

This hybrid types of metaheuristics are motivated by the QAP problems. For the QAP using a neighborhood based on the exchange operator, the cost of exploring local search is quadratic with the size of the neighbourhood. Thus, it is computational inefficient to use neighbourhood functions of large size like required by VNS algorithms. The computational cost of the mutation operator, however, does not change much with the exchange rate. This unbalance between the computational cost of the two techniques for the QAP problem makes the study of the alternation between the two techniques relevant for this class of problems.

3.1 A Baseline Algorithm

In this setting both restarting strategies can be used simultaneously. The pseudo-code for this algorithm is given in Algorithm 3.

Algorithm 3. Multi - operator MetaHeuristics (MMH)

```
$t \leftarrow 0$
$s' \leftarrow$ GenerateInitialSolution
$s^{(0)} \leftarrow$ LocalSearch$(s', (\mathcal{N}_1, \ldots, \mathcal{N}_P))$
while The stopping criteria is NOT met do
    $s' \leftarrow$ Perturbation$(s^{(t)})$
    $s^{(t+1)} \leftarrow$ LocalSearch$(s', (\mathcal{N}_1, \ldots, \mathcal{N}_P))$
    $t \leftarrow t + 1$
end while
return The local optimum solution $s^{(t)}$
```

At the initialization, a solution s' is generated uniform randomly with GenerateInitialSolution. The local search function LocalSearch is initialized with the initial solution s' and with the set of neighbourhood strategies $(\mathcal{N}_1, \ldots, \mathcal{N}_P)$. The solution returned by local search is an initial local optimum $s^{(0)}$.

Each iteration, until a stopping criteria is met, a new restarting solution s' is proposed with a perturbator operator over the current local optimum solution $s^{(t)}$. The local search is restarted from this newly generated solution s' and using the set of neighbourhood functions $\mathcal{N}$. The new local optimum $s^{(t+1)}$ improves or it is equal with the current local optimum $s^{(t)}$. The counter of iterations is updated $t \leftarrow t + 1$.

Remark 3.1. We assume here a small number of neighbourhoods in the neighbourhood set $\mathcal{N}$. Then, the local optimum resulted from the local search function is probable not the global optimum and the search needs to be restarted from another region of the search space.

Remark 3.2. The m-exchange rate of the mutation operator needs to be larger than the largest neighbourhood from the neighbourhood set in order to escape the basin of attraction of the current local optimum. A smaller m-exchange rate still make sense with the first improvement strategy where the search is stopped at the first improvement.

Remark 3.3. If the perturbator operator is a uniform random generator, then we have a skewed VNS because the local search is uniform randomly restarted in the search space. If there is only one neighbourhood function in the neighbourhood set, then we have a standard ILS search like in Algorithm 1.

3.2 Selecting One Restarting Strategy

The baseline algorithm can be quite computational demanding especially for the QAP problem where the size of the neighbourhoods increase quadratically. We propose a variant of the multi-restart strategies metaheuristics where the restarting techniques are alternated rather then simultaneously used.

At the initialization, an uniform random solution is generated s' and a neighbourhood strategy is uniform randomly selected $\mathcal{N}^{(0)}$. The local search LocalSearch$(s', \mathcal{N}^{(0)})$ returns the first local optimum.

Algorithm 4. Alternating operators MetaHeuristics (AMH)

```
$t \leftarrow 0$
$s' \leftarrow$ GenerateInitialSolution
$\mathcal{N}^{(0)} \leftarrow$ SelectInitialNeighbourhood($\mathcal{N}_1, \ldots, \mathcal{N}_P$)
$s^{(0)} \leftarrow$ LocalSearch($s', \mathcal{N}^{(0)}$)
while the stopping criteria is NOT met do
    if SelectRestartStategy is mutation then
        $s' \leftarrow$ Perturbation($s^{(t)}$)
        $s^{(t+1)} \leftarrow$ LocalSearch($s', \mathcal{N}_1$)
    else
        $\mathcal{N}^{(t)} \leftarrow$ SelectNeighbourhood($\mathcal{N}_1, \ldots, \mathcal{N}_P$)
        $s^{(t+1)} \leftarrow$ LocalSearch($s^{(t)}, \mathcal{N}^{(t)}$)
    end if
    $t \leftarrow t + 1$
end while
return The local optimum solution $s^{(t)}$
```

Each iteration, until a stopping criteria is met, a restarting strategy is chosen using a function SelectRestartStategy. This strategy can be simply a uniform generator, and then the two restart strategies are uniform randomly selected.

If the selected strategy is the mutation operator, then a (probable suboptimal) solution is generated with the perturbator function in order to restart local search. Like in ILS, the local search will now use a small neighbourhood, where we assume that $\mathcal{N}_1$ is the smallest neighbourhood from the set $\mathcal{N}$. If the variable neighbourhood search is selected then only the neighbourhood is selected with SelectNeighbourhood and the local search is restarted from the current local optimal solution $s^{(t)}$.

Remark 3.4. If only one of the restarting strategies is used, this algorithm is a standard ILS or a standard VNS, after case.

3.3 Setting-Up the Parameters

The two restart strategies have two operators whose parameters need to be set: i) the mutation operator, and ii) the set of neighbourhood functions. For each operator, we consider a fixed set of parameters.

The Set of Mutation Parameters. For the mutation operator, we consider K parameters, where $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_K)$. A mutation operator that uniform randomly select each parameter from the set $\mathcal{M}$ is shown to outperform the ILS algorithms that use a mutation operator with a single parameter from $\mathcal{M}$.

For the QAP instances, we consider each mutation exchange rate to be a mutation parameter.

Adaptive Operator Selection. We correlate the usage of a specific mutation operator with the improvement resulted by applying this mutation operator on the current solution. The adaptive pursuit (AP) strategy [7] is often used for adaptive operator selection.

The mutation operator for which the restarted local search improves the most the current solution, will be pursuit often.

The Set of Neighbourhood Function. We assume a fixed set of P neighbourhood functions, $\mathcal{N} = (\mathcal{N}_1, \ldots, \mathcal{N}_P)$. The problem of selecting one neighbourhood function from a set of neighbourhood functions is similar to selecting a perturbator operator from a set of perturbator operators. Thus, we could uniform randomly select these operators or we could adaptively select the neighbourhoods that improve the most the current solution.

Remark 3.5. In general, in Evolutionary Computation, the parameters of different operators are selected independently. For our problem, the parameters from the mutation set are independently selected from the parameters of the neighbourhood set.

In the next section, we propose a new adaptive operator selection algorithm that consider that the performance of the two operators is correlated and thus that their selection mechanisms should be correlated.

4 Adaptive Multi-operator MetaHeuristics

In this section, we consider an adaptive version of the multi-operator metaheuristics that includes an extension of the adaptive pursuit strategy [7,12] for multi-operator selection. The two operators can be adapted separately, independently or simultaneously.

The standard *adaptive pursuit* (AP) algorithm adapts a probability vector $\mathcal{P}^{(t)}$ such that the operator that has the highest estimated reward is chosen with very high probability. The target distribution does not change in time and the target distribution is associated with a step-like distribution where one operator has a very large selection probability whereas the probability of selecting the rest of the operators is much smaller.

The biggest difference between the single and the multi-operator adaptive pursuit is that now the rewards are vectors, $\mathbf{R}^{(t)}$, instead of values. Consequently, the temporal probabilities vectors, i.e. $\mathbf{P}^{(t)}$ and $\mathbf{Q}^{(t)}$, are also matrices. Consider the mutation and the neighbourhood operators, each operator with K and P parameters as before. The probability distribution for the mutation operator and for the neighbourhood operator is

$$\mathbf{P}^{(t)}_{\mathcal{N}} = \{\mathbf{P}^{(t)}_{11}, \ldots, \mathbf{P}^{(t)}_{1K}\}, \qquad \mathbf{P}^{(t)}_{\mathcal{M}} = \{\mathbf{P}^{(t)}_{21}, \ldots, \mathbf{P}^{(t)}_{2P}\}$$

where $\forall t$, and $\forall j$, and $0 \leq \mathcal{P}^{(t)}_{1j} \leq 1$, $\sum_{j=1}^{K} \mathcal{P}^{(t)}_{1j} = 1$, and $\forall t$, and $\forall j$, and $0 \leq \mathcal{P}^{(t)}_{2j} \leq 1$, $\sum_{j=1}^{P} \mathcal{P}^{(t)}_{1j} = 1$. The quality distribution (or the estimated reward) for the mutation operator and for the neighbourhood operator is

$$\mathbf{Q}^{(t)}_{\mathcal{N}} = \{\mathbf{Q}^{(t)}_{11}, \ldots, \mathbf{Q}^{(t)}_{1K}\}, \qquad \mathbf{Q}^{(t)}_{\mathcal{M}} = \{\mathbf{Q}^{(t)}_{21}, \ldots, \mathbf{Q}^{(t)}_{2P}\}$$

4.1 A Baseline Algorithm

We consider the multi-operator adaptive pursuit algorithm where the probability of selecting a parameter is independently adapted for each operator. Algorithm 5 represents the pseudo-code of this algorithm.

Algorithm 5. Adaptive Multi - operator MetaHeuristics (AMMH)

```
$t \leftarrow 0$
$(\mathbf{P}^{(0)}, \mathbf{Q}^{(0)}) \leftarrow$ InitializeAdaptationVectors
$s' \leftarrow$ GenerateInitialSolution
$s^{(0)} \leftarrow$ LocalSearch($s', (\mathcal{N}_1, \ldots, \mathcal{N}_P)$)
while The stopping criteria is NOT met do
    $(v_\mathcal{M}, v_\mathcal{N}) \leftarrow$ SelectParameters($\mathbf{P}^{(t)}$)
    $s' \leftarrow$ Perturbation($s^{(t)}, \mathcal{M}, v_\mathcal{M}$)
    $\mathcal{N}^{(t)} \leftarrow$ SelectNeighbourhood($\mathcal{N}, v_\mathcal{N}$)
    $s^{(t+1)} \leftarrow$ LocalSearch($s^{(t)}, \mathcal{N}^{(t)}$)
    $\mathbf{R}_\mathbf{v}^{(t)} \leftarrow$ UpdateRewardVector($v_\mathcal{M}, v_\mathcal{N}$)

    Update $\mathbf{Q}^{(t+1)}$ with reward $\mathbf{R}_\mathbf{v}^{(t)}$
    $\mathbf{r} \leftarrow$ HighRankQ($\mathbf{Q}^{(t+1)}$)
    $\mathbf{P}^{(t+1)} \leftarrow$ UpdateProbability($\mathbf{P}^{(t)}, \mathcal{D}_\mathbf{r}, \beta$)

    $t \leftarrow t + 1$
end while
return The local optimum solution $s^{(t)}$
```

InitializeAdaptationVectors. In the initialization step, $t \leftarrow 0$ and the vectors of each parameter are initialized separately. For the mutation operator, we have $\mathcal{P}_{1j}^{(t)} \leftarrow 1/K$ such that $\sum_{i=1}^{K} \mathcal{P}_{1j}^{(t)} = 1$. For all j, $1 \leq j \leq K$, the quality values are equal $\mathcal{Q}_{1j}^{(t)} \leftarrow 0.5$ meaning that all the parameters are considered equally important at the initialization. In the sequel, for the neighbourhood operator, we have $\mathcal{P}_{2i}^{(t)} \leftarrow 1/P$ and $\mathcal{Q}_{2i}^{(t)} \leftarrow 0.5$, where $\forall i$, $1 \leq i \leq P$.

SelectParameters independently selects parameters for each operator. For the mutation operator, an operator $v_\mathcal{M} \in \mathcal{M}$ is selected proportionally with the corresponding probability distribution $\mathbf{P}_\mathcal{M}^{(t)} \leftarrow (\mathcal{P}_{11}^{(t)}, \ldots, \mathcal{P}_{1K}^{(t)})$. For the neighbourhood operator, an operator $v_\mathcal{N} \in \mathcal{N}$ is selected proportionally with the probability distribution $\mathbf{P}_\mathcal{N}^{(t)} \leftarrow (\mathcal{P}_{21}^{(t)}, \ldots, \mathcal{P}_{2P}^{(t)})$. The resulting parameter vector is denoted with $\mathbf{v} \leftarrow (v_\mathcal{M}, v_\mathcal{N})$.

UpdateRewardVector. An improvement in the cost of the candidate solution $s^{(t+1)}$ when compared with the cost of the current solution $s^{(t)}$ means that $c(s^{(t+1)}) < c(s^{(t)})$. The reward $\mathcal{R}_\mathbf{v}^{(t)}$ for using the vector of parameters $\mathbf{v}$ in restarting the local search is connected with the improvement over the current solution

$$\mathcal{Q}_\mathcal{M}^{(t)} = \frac{\#\ improv\ of\ v_\mathcal{M}}{\#\ trials\ of\ v_\mathcal{M}}, \qquad \mathcal{Q}_\mathcal{N}^{(t)} = \frac{\#\ improv\ of\ v_\mathcal{N}}{\#\ trials\ of\ v_\mathcal{N}}$$

In the initialization step, we ensure that all the operators are tried at least once. If applying $v_\mathcal{M}$ results in an improvement, $\mathcal{Q}_\mathcal{M}^{(t)}$ is increasing

$$\mathcal{Q}_\mathcal{M}^{(t)} < \mathcal{Q}_\mathcal{M}^{(t+1)} \Leftrightarrow \frac{\#\ improv\ v_\mathcal{M}}{\#\ total\ v_\mathcal{M}} < \frac{\#\ improv\ v_\mathcal{M} + 1}{\#\ total\ v_\mathcal{M} + 1}$$

Otherwise, $\mathcal{Q}_{\mathcal{M}}^{(t)}$ decreases

$$\mathcal{Q}_{\mathcal{M}}^{(t)} > \mathcal{Q}_{\mathcal{M}}^{(t+1)} \Leftrightarrow \frac{\#\ improv\ v_{\mathcal{M}}}{\#\ total\ v_{\mathcal{M}}} > \frac{\#\ improv\ v_{\mathcal{M}}}{\#\ total\ v_{\mathcal{M}}+1}$$

Similar properties we have for $\mathcal{Q}_{\mathcal{N}}^{(t)}$.

Rank Quality Vectors. The function call HighRankQ independently high ranks the quality vectors for each operator $\mathbf{Q}_{\mathcal{M}}^{(t+1)}$ and $\mathbf{Q}_{\mathcal{N}}^{(t+1)}$. There is a quality ranking matrix $\mathbf{r}$ that has an independent quality vector for each operator, $\mathbf{r} = (\mathbf{r}_{\mathcal{M}}, \mathbf{r}_{\mathcal{N}})$. Thus, an instance of a quality vector $\mathbf{r}$ can be highly ranked for one operator and lower ranked for the other operator.

Update Probabilities. The two probability distributions of $\mathbf{P}^{(t)}$ are also independently updated for each operator. For all j, $1 \leq j \leq K$, we have that

$$\mathcal{P}_{1j}^{(t+1)} \leftarrow \mathcal{P}_{1j}^{(t)} + \beta \cdot (\mathcal{D}_{r_{1j}} - \mathcal{P}_{1j}^{(t)})$$

and for all i, $1 \leq i \leq P$, we have that

$$\mathcal{P}_{2i}^{(t+1)} \leftarrow \mathcal{P}_{2i}^{(t)} + \beta \cdot (\mathcal{D}_{r_{2i}} - \mathcal{P}_{2i}^{(t)})$$

The target distribution, $\mathbf{D}$, is a step-like distribution for each operator. Thus,

$$\mathcal{D}_{\mathcal{M}} = [p_M, \underbrace{p_m, \ldots, p_m}_{K-1}], \text{ where } p_M = 1 - (K-1) * p_m$$

and

$$\mathcal{D}_{\mathcal{N}} = [p_N, \underbrace{p_n, \ldots, p_n}_{P-1}], \text{ where } p_N = 1 - (P-1) * p_n$$

For each operator, only one element has the maximum value p_M or p_N and the rest of the operators are updated with a low probability, p_m or p_n, after case. In updating the j-th element in $\mathcal{P}_{\mathcal{M}}^{(t)}$, the rank j in $\mathcal{D}_{\mathcal{M}}$ is used. If $\mathcal{D}_{\mathcal{M}}$ is a valid probability vector, i.e., $\sum_{j=1}^{K} \mathcal{D}_{1j} = 1$, then the elements in the probability vector $\mathcal{P}_{\mathcal{M}}^{(t)}$ sum up to 1. In the sequel, if $\mathcal{D}_{\mathcal{N}}$ is a valid probability vector, i.e., $\sum_{i=1}^{P} \mathcal{D}_{2i} = 1$, then the elements in the probability vector $\mathcal{P}_{\mathcal{N}}^{(t)}$ sum up to 1.

The Algorithm. The adaptive multi-operator metaheuristics algorithm starts by initializing the adaptation probability distributions. The initial solution s' and the initial neighbourhood $\mathcal{N}^{(0)}$ are used to restart local search.

Each iteration, two parameters, $v_{\mathcal{M}}$ and $v_{\mathcal{N}}$, which is one parameter for each operator, are generated using the function SelectParameters. A new solution s' is generated using the perturbator function over the current solution $s^{(t)}$, and the given mutation parameter $v_{\mathcal{M}}$. A neighbourhood function $\mathcal{N}^{(t)}$ is selected from the set of neighbourhoods $\mathcal{N}$ as indicated by the parameter $v_{\mathcal{N}}$. The local search is restarted from solution s' using the neighbourhood $\mathcal{N}^{(t)}$. The reward vector is updated and the quality vectors are ranked in order to select the best quality vector. The probability distribution is updated.

Remark 4.1. Adaptive pursuit has two extra parameters that could be tuned: i) minimum (and maximum) selection probabilities, and ii) the learning rate β. Let $\beta \in [0,1]$ be the learning rate that determines the speed with which the algorithm converges to the maximum and minimum estimated reward values. A large β value means faster convergence of the algorithm to the target probabilities, which means a poorer use of certain operators. A small β value means slower convergence and thus more chances for the less rewarded operators to be tested.

Remark 4.2. From the three proposed multi-operator metaheuristics, Algorithm 5 and Algorithm 3 resemble the most since both algorithms will simultaneously use both operators to restart the local search, whereas Algorithm 4 alternatively uses the two operators.

5 Experimental Results

In this section, we show preliminary experimental results that compare the performance of seven metaheuristics on two QAP instances.

Tested Metaheuristics Algorithms. We compare the performance of the following metaheuristics:

MLS the multi-restarted local search algorithm uniform randomly restarts LS;
ILS the iterated local search algorithm uses a set of mutation exchange rates $\mathcal{M}$;
ALS the adaptive iterated local search algorithm uses adaptive pursuit on a set of mutation exchange rates $\mathcal{M}$;
VNS the variable neighbourhood search algorithm uses a set of neighbourhood functions $\mathcal{N}$;
MMH the multi-operator metaheuristic algorithm uses simultaneously a set of mutation exchange rates $\mathcal{M}$ and a set the neighbourhood functions $\mathcal{N}$;
AMH the alternating operators metaheuristic algorithm alternates the restarting operators;
AMMH the adaptive multi-operator metaheuristic algorithm adaptively selects the parameters for mutation and the neighbourhood function.

Note that four algorithms can be classified as iterated local search algorithms, *MLS*, *ILS*, *ALS* and *AMMH*. Four of the above algorithms can be classified as variable neighbourhood search algorithms, *VNS*, *MMH*, *AMH* and *AMMH*. The *AMMH* algorithm can be classified both as ILS and VNS since it uses both type operators to restart LS.

Tested QAP Instances. We use composite QAP (cQAP) instances [9] to test the seven metaheuristic algorithms. These QAP instances are automatically generated such that they have the following properties: i) large size, ii) difficult and interesting to solve with both heuristics and exact algorithms, iii) known optimum solution, and iv) not trivial asymptotic behaviour (i.e., the difference between the lower and the upper cost functions does not go to 0 when the number of facilities goes to ∞).

For our experiments, we consider two cQAP instances with a medium number of facilities, $N = \{20, 32\}$. The optimum solution is always the identity permutation.

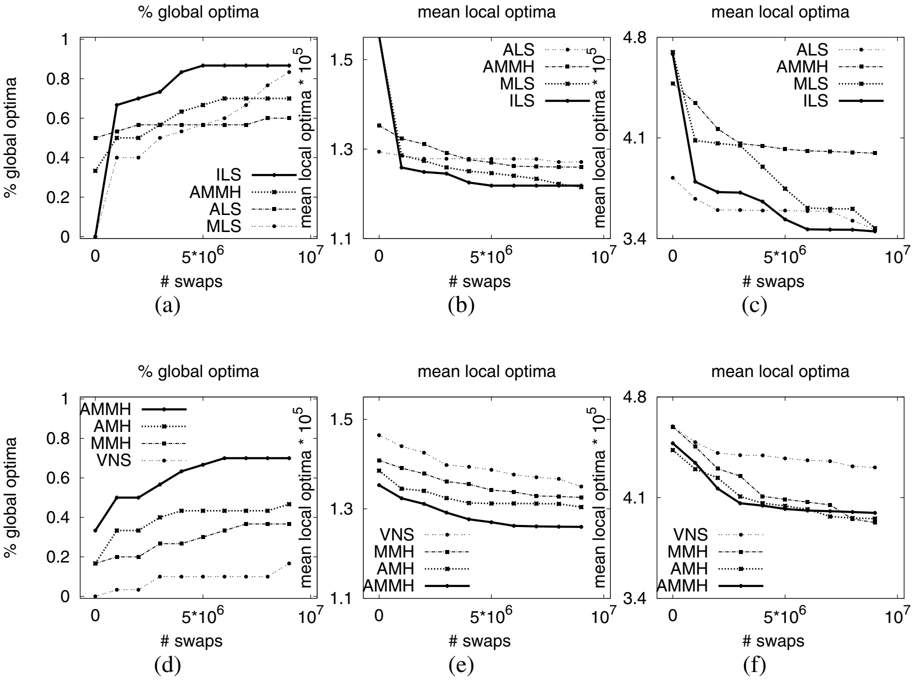


Fig. 1. (On the left) The percent of runs in which of the local optima value for cQAP with $N = 20$ is found for the seven algorithms. (In the middle) The mean cost function of the seven algorithms when $N = 20$, and (on the right) the mean cost function when $N = 32$. (On the top) The four iterative based metaheuristic algorithms, and (on the bottom) the four variable neighbourhood search based metaheuristic algorithms.

Setting the Parameters. The set of the mutation exchange rates is set to $\mathcal{M} = \{3, 4, 5, 6, 7\}$. Thus, $K = |\mathcal{M}| = 5$. The set of neighbourhood functions cannot be very large for the QAP instances because the exploration of their neighbourhood increases quadratically with the exchange rates $\mathcal{N} = \{2, 3, 4\}$. Thus, $P = |\mathcal{N}| = 3$.

Each metaheuristic algorithm is run for 100 times for an equal amount of exchanges 10^6.

For the multi-operator adaptive pursuit, we consider a standard learning rate of $\beta = 0.01$. We set the maximum and the minimum probability distribution such that the proportion between these probabilities is 6. Then, for the mutation operator we have $p_M = 0.6$ and $p_m = 0.1$, and for the neighbourhood operator we have $p_N = 0.7$ and $p_n = 0.15$.

5.1 Results

In Figure 1, to compare the performance of the discussed metaheuristics, we measure the percentage of times the global optima is found. For $N = 32$, the number of identified local optima is 0 for all the algorithms showing the increase in complexity of the problem with the increase in number of facilities.

$N = 20$**.** The best algorithm is the iterated local search (*ILS*) followed by multi-restart local search (*MLS*). The worst algorithm is the variable neighbourhood search algorithm (*VNS*). In general, the meta-heuristics algorithms that use mutation perturbators, in Figure 1 on the left, outperform the algorithms that alternates the neighbourhood functions, in Figure 1 on the right. The adaptive multi-operator metaheuristics (*AMMH*) has a better performance than both the multi-operator metaheuristics (*MMH*) and the adaptive iterated local search (*ALS*).

$N = 32$**.** For larger cQAP instances, the adaptive iterated local search (*ALS*) is performing similarly with the iterative local search (*ILS*). The adaptive multi-operator metaheuristics (*AMMH*) is again the best algorithm that uses variable neighbourhood search, but this time is outperformed by the adaptive iterated local search (*ALS*).

Discussion. We conclude that for smaller size cQAPs, adaptation of the neighbourhood function is more beneficial than the adaptation of the mutation operators, whereas for larger size cQAPs, this situation is vice-versa. This could indicate that the the usage of multi restarting operators can be beneficial in certain cases.

6 Conclusions

We have proposed a new class of metaheuristics that uses two operators to restart the search, i.e. the stochastic perturbator operators and the neighbourhood function. We motivate the multi-operator metaheuristic algorithms with the quadratic assignment problem where the size of the neighbourhood increases quadratically with the exchange rate, whereas the generation of a solution with mutation increases only linearly.

We propose two instances of multi-operator metaheuristic algorithms. One instance uses the two restarting techniques simultaneously, but the alternation of the neighbourhood functions is the most important procedure in the local search call. The second multi-operator metaheuristics algorithm uses the two operators alternatively: i) the mutation operator is used with the small neighbourhood function, and ii) the local search is now using a single, uniformly selected, neighbourhood function. We consider the last algorithm to be more balanced in using the two restarting operators than the first algorithm. The third proposed algorithm is an adaptive multi-operator metaheuristic algorithm where the adaptive pursuit is extended to a multi-objective adaptive pursuit. In this variant of multi-operator adaptive pursuit, both operators are adapted separately.

Preliminary experimental results compare the performance of the discussed metaheuristics and show a better performance of the multi-operator metaheuristics as compared with the variable neighbourhood search metaheuristics for the tested QAP instances. An interesting perspective is to apply the same methodology to solve multi-objective combinatorial optimization problems.

References

[1] Talbi, E.G.: Metaheuristics: from design to implementation. Wiley (2009)

[2] Hoos, H.H., Stutzle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann (2005)

[3] Mladenovic, N., Hansen, P.: Variable neighborhood search. Computers & Operations Research 24(11), 1097–1100 (1997)

[4] Hansen, P., Mladenovic, N., Perez, J.A.M.: Variable neighbourhood search: methods and applications. 4OR 6(4), 319–360 (2008)

[5] Drugan, M.M., Thierens, D.: Path-Guided Mutation for Stochastic Pareto Local Search Algorithms. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 485–495. Springer, Heidelberg (2010)

[6] Drugan, M.M., Thierens, D.: Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies. J. Heuristics 18(5), 727–766 (2012)

[7] Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: Proc of Conf. of Genetic and Evol. Comp. (GECCO 2005), pp. 1539–1546 (2005)

[8] Loiola, E., de Abreu, N., Boaventura-Netto, P., Hahn, P., Querido, T.: A survey for the quadratic assignment problem. European J. of Operational Research 176(2), 657–690 (2007)

[9] Drugan, M.M.: Generating QAP instances with known optimum solution and additively decomposable cost function. Jornal of Combinatorial Optimization (to appear, 2014)

[10] Stützle, T.: Iterated local search for the quadratic assignment problem. European Journal of Operational Research 174(3), 1519–1539 (2006)

[11] Hansen, P., Mladenovic, N.: First vs. best improvement: An empirical study. Discrete Applied Mathematics 154(5), 802–817 (2006)

[12] Drugan, M.M., Thierens, D.: Generalized adaptive pursuit algorithm for genetic Pareto local search algorithms. In: Proc. of Conf. of Genetic and Evol. Comp. (GECCO 2011), pp. 1963–1970 (2011)

A New Predictor Corrector Variant for Unconstrained Bi-objective Optimization Problems

Adanay Martín and Oliver Schütze

Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, C.P. 07360,
Col. San Pedro Zacatenco, Mexico City, México
amartin@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx

Abstract. Many real-life applications can be formulated as a multi-objective optimization problem. Since the solutions set, the Pareto set, of such problems typically forms locally a manifold, specialized predictor-corrector methods that allow to follow the solution set from a given optimal point have proven to be very effective for such problems.

In this work, we investigate two different possible choices for the predictor direction for bi-objective problems: One that was recently proposed in literature and another one which is a variant of the 'classical' choice adapted to multi-objective optimization but has the advantage that it does not require the consideration of the weight space coming from the Karush-Kuhn-Tucker formulation in an augmented system. From our observations we derive three new continuation variants and compare them on some benchmark models. The numerical results indicate that the novel choice of the predictor that uses the Hessians of the objectives leads to some savings in the computational effort compared to the classical continuation method, and that its Hessian free realization via Quasi-Newton methods leads to further significant improvements in the overall cost in terms of function evaluations.

Keywords: multi-objective optimization, continuation methods, Newton method, Quasi-Newton method.

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). As a general example, two common goals in product design are certainly to maximize the quality of the product *and* to minimize its cost. Since these two goals are typically contradictory, it comes as no surprise that the solution set—the so-called *Pareto set*—of a MOP does in general not consist of one single solution but rather of an entire set of solutions. In fact, the Pareto set typically forms a $(k-1)$-dimensional object, where k is the number of objectives involved in the MOP ([1]).

In literature, a huge variety of methods dealing with the approximation of the entire Pareto set, respectively its image, the Pareto front, can be found. There are, for instance, many scalarization methods which transform the MOP into a 'classical' scalar optimization problem (SOP). By choosing a clever sequence of SOPs a suitable finite

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_12, © Springer International Publishing Switzerland 2014

size approximation of the entire Pareto set can be obtained (see [2–5] and references therein). Further, there exist many set oriented methods such as specialized evolutionary strategies ([6, 7]), subdivision techniques ([8–10]) or cell mapping techniques ([11, 12]). These methods have in common that they use sets in an iterative manner and are thus able to deliver an approximation of the solution set in one run of the algorithm.

Since the Pareto set forms under some mild regularity conditions locally a $(k-1)$-manifold, specialized continuation methods which perform a search along the Pareto set are very efficient if one (or more) solution is at hand. Such algorithms can for instance be found in [1, 13–17].

In this paper, we discuss further continuation methods for bi-objective problems (i.e., $k = 2$). Since so far many numerical methods for the detection of Pareto optimal solutions exist—next to the above mentioned there are the steepest descent method ([18]), specialized Newton ([19]) and Quasi-Newton ([20]) methods—we focus here on the choice of the predictor. In particular we will discuss two choices:

First, we will investigate the use of the gradient at a given Pareto point as e.g. done in [17]. We will show that this choice has the advantage that the most greedy search in objective space can be expected. On the other hand, the gradient does not have to point along the linearized Pareto set which results in a potential loss of efficiency for more complicated models.

Further, we will investigate another choice that is a variant of the classical predictor as e.g. considered in [1, 13]. This variant, however, will have the advantages that (i) the weight space can be considered separately which does not augment the problem and (ii) it can be made ‘Hessian free’ via utilizing a Quasi-Newton method (e.g., [21]).

Based on these predictors, we will investigate several possible predictor-corrector methods and investigate their performances numerically on some examples. First results indicate that the choice of the second predictor discussed here leads to an increase of the performance.

The remainder of this paper is organized as follows: In Section 2, we briefly present the background required for the understanding of this paper. In Section 3, we discuss two possible predictor choices. The predictor-corrector variants as well as numerical results on some benchmark functions are presented in Section 4. Finally, we make some conclusions and draw possible paths for future research in Section 5.

2 Background

Here we briefly state the basic facts about multi-objective optimization and summarize the main steps performed in ‘classical’ Predictor-Corrector (PC) algorithms as e.g. discussed in [13].

2.1 Multi-Objective Optimization

In the following we consider unconstrained continuous MOPs

$$\min_{x\in\mathbb{R}^n}\{F(x)\}, \tag{MOP}$$

where F is defined as the vector of the objective functions $F : \mathbb{R}^n \to \mathbb{R}^k$, $F(x) = (f_1(x), \ldots, f_k(x))$, and where each objective $f_i : \mathbb{R}^n \to \mathbb{R}$ is (for simplicity) sufficiently smooth. For $k = 2$, the above problem is called a bi-objective problem (BOP).

The optimality of a MOP is defined by the concept of *dominance* ([22]): A vector $y \in \mathbb{R}^n$ is *dominated* by a vector $x \in \mathbb{R}^n$ ($x \prec y$) with respect to (MOP) if $f_i(x) \leq f_i(y)$, $i = 1, \ldots, k$, and there exists an index j such that $f_j(x) < f_j(y)$, else y is non-dominated by x. A point $x \in \mathbb{R}^n$ is called *(Pareto) optimal* or a *Pareto point* if there is no $y \in \mathbb{R}^n$ which dominates x. The set of all Pareto optimal solutions is called the *Pareto set*, and is denoted by $\mathcal{P}$. The image $F(\mathcal{P})$ of the Pareto set is called the *Pareto front*. Both sets typically form a $(k-1)$-dimensional object.

2.2 Predictor Corrector Methods

Assume we are given an implicitly defined problem

$$H(x) = 0, \tag{1}$$

where $H : \mathbb{R}^{N+1} \to \mathbb{R}^N$ is sufficiently smooth. If x is a solution of (1) with $rank(H(x)) = N$, then it follows by the Implicit Function Theorem that there exist a value $\epsilon > 0$ and a curve $c : (-\epsilon, \epsilon) \to \mathbb{R}^{N+1}$ such that $c(0) = x$ and

$$H(c(s)) = 0, \quad \forall s \in (-\epsilon, \epsilon) \tag{2}$$

Differentiating (2) leads to

$$H'(c(s)) \cdot c'(s) = 0. \tag{3}$$

Hence, tangent vectors $c'(s)$ (and thus, linearizations of the solution curve at $x = c(s)$) can be found via computing kernel vectors of $H'(x)$. This is done in literature via a QR factorization of $H'(x)^T$: If

$$H'(x)^T = QR \tag{4}$$

for an orthogonal matrix $Q \in \mathbb{R}^{(N+1)\times(N+1)}$ and a right upper matrix $R \in \mathbb{R}^{(N+1)\times N}$, then the last column vector q_{N+1} of Q is such a desired kernel vector. The orientation of the curve (note that both $+q_{N+1}$ and $-q_{N+1}$ are desired linearizations, but point in different directions) can be computed by monitoring the sign of

$$\det \begin{pmatrix} H'(x) \\ q_{N+1}^T \end{pmatrix}. \tag{5}$$

Having computed the vector pointing along the linearized solution curve, a line search in that direction can be performed leading to the predictor solution p. In a following corrector step, one can get back to c by using (1) together with a Gauss-Newton or Levenberg-Marquardt method ([21]) starting with p.

3 On the Choice of the Predictor for BOPs

In the following we discuss two possible choices for the predictor direction within a PC method for the treatment of BOPs: The gradient of one of the objectives which is e.g. used in [17] as well as one variant of the one presented in Section 2.2 which is formulated using the Hessians of each objective, but which has some interesting properties.

Assume we are given a Pareto point of a bi-objective problem of the form (MOP) with KKT multipliers $\alpha \in \mathbb{R}^2$, i.e., $\alpha_1, \alpha_2 \geq 0$ with $\alpha_1 + \alpha_2 = 1$ and

$$\alpha_1 \nabla f_1(x) + \alpha_2 \nabla f_2(x) = J(x)^T \alpha = 0, \tag{6}$$

where

$$J(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \nabla f_2(x)^T \end{pmatrix} \in \mathbb{R}^{2 \times n}, \tag{7}$$

denotes the Jacobian of F at x. Further, we assume that $\alpha_i \neq 0$, $i = 1, 2$, as well as $\nabla f_i(x) \neq 0, i = 1, 2$.

One possible choice for the predictor direction is the gradient of one of the objectives at x, i.e.,

$$\nu_g := \pm \frac{\nabla f_1(x)}{\|\nabla f_1(x)\|_2} \tag{8}$$

as e.g. done in [17]. One appealing fact is that a movement in ν_g via line search (i.e., $x_{new} = x + t\nu_g$ for a step size $t \in \mathbb{R}_+$) yields the maximal movement along the linearized Pareto front at $F(x)$ in objective space as the following discussion shows:
It is known that the KKT multipliers α is orthogonal to the linearized Pareto front at $F(x)$ ([1]). Further, for a direction $\nu \in \mathbb{R}^n$ in parameter space the corresponding movement in objective space for infinitesimal step sizes is given by

$$d_\nu := J(x)\nu \tag{9}$$

To see this, consider the i-th component of the right hand side of (9):

$$(J(x)\nu)_i = \lim_{t \searrow 0} \frac{f_i(x + t\nu) - f_i(x)}{t} = \langle \nabla f_i(x), \nu \rangle, \quad i = 1, \ldots, k. \tag{10}$$

Since x is a Pareto point, almost all values of ν lead to a movement along the linearized Pareto front:

$$\langle J(x)\nu, \alpha \rangle = \langle \nu, J(x)^T \alpha \rangle \overset{(6)}{=} 0. \tag{11}$$

It is $J(x)^T \nu = 0$ if and only if ν is orthogonal to one of the objectives' gradients. Now let $\nu_g = \nabla f_1(x)/\|\nabla f_1(x)\|_2$ (analog for the opposite direction). Then we have

$$\nabla f_2(x) = \lambda \nabla f_1(x), \tag{12}$$

where $\lambda = -\alpha_1/\alpha_2 < 0$. Then it is

$$\|J(x)\nu\|_2^2 = (1 + \lambda^2)\langle \nabla f_1(x), \nu \rangle^2 \tag{13}$$

It is known that the maximal value in (13) on $\{v : \|\nu\|_2 = 1\}$ is obtained for $\nu = \nu_g$, and the above claim follows.

Hence, ν_g can be considered as the most promising direction vector for movements along the Pareto front which is certainly appealing. On the other hand, movements along

ν_g do *not* perform a movement along the linearized Pareto *set* at x. The choice of ν_g might thus lead to a potential loss in the performance for particular BOPs. To underline the last statement consider the BOP $F : \mathbb{R}^2 \to \mathbb{R}^2$:

$$\begin{aligned} f_1(x) &= (x_1 - 1)^4 + (x_2 - 1)^2 \\ f_2(x) &= (x_1 + 1)^2 + (x_2 + 1)^4 \end{aligned} \tag{14}$$

Figure 1 shows the Pareto set that forms a curve connecting the points $(-1, -1)^T$ and $(1, 1)^T$ as well as the gradient directions ν_g along the set. In this example, the objectives' gradients point almost orthogonal to the linearized Pareto set near to its end points.

In the following we describe our second proposal for the predictor which is a variant of the one used in [1] but which seems to be more promising for the use within Pareto continuation. For this, consider a general unconstrained MOP (i.e., a general number of objectives). A function $\tilde{F}$ such that the Pareto set is contained in $\tilde{F}^{-1}(0)$ is given by ([1]):

$$\begin{aligned} &\tilde{F} : \mathbb{R}^{n+k} \to \mathbb{R}^{n+1} \\ &\tilde{F}(x, \alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix} = 0 \end{aligned} \tag{15}$$

The map $\tilde{F}$ is motivated by the Karush-Kuhn-Tucker equations and is defined in (x, α)-space. As for (3) we are interested in kernel vectors of $\tilde{F}'(x, \alpha)$ for a given Pareto point

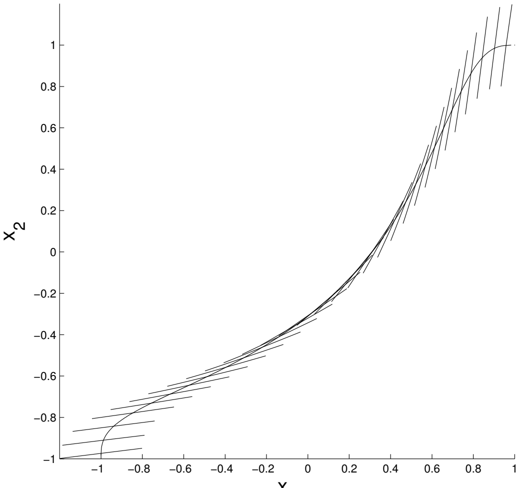


Fig. 1. Directions of the objectives' gradients along the Pareto front of MOP (14)

with associated weight α. One way is to use a QR factorization as in (4), another one is to do it directly as follows: Let $\nu \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^k$ such that

$$\tilde{F}'(x,\alpha)\begin{pmatrix}\nu\\ \mu\end{pmatrix} = \begin{pmatrix}\sum_{i=1}^k \alpha_i \nabla^2 f_i(x) & \nabla f_1(x) & \dots & \nabla f_k(x)\\ 0 & 1 & \dots & 1\end{pmatrix}\begin{pmatrix}\nu\\ \mu\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix} \tag{16}$$

By the second equation in (16) we see that

$$\sum_{i=1}^k \mu_i = 0 \tag{17}$$

Assume for now that we are given $\mu \neq 0$ such that the above property if fulfilled then we obtain by the first equation in (16)

$$\sum_{i=1}^k \alpha_i \nabla^2 f_i(x)\nu = -\sum_{i=1}^k \mu_i \nabla f_i(x) = -J(x)^T\mu \tag{18}$$

If for instance all objectives of F are strictly convex it follows that the matrix

$$H_\alpha := \sum_{i=1}^k \alpha_i \nabla^2 f_i(x) \in \mathbb{R}^{n\times n} \tag{19}$$

is positive definite (and thus invertible), and for the tangent vector it holds

$$\nu_\mu = -H_\alpha^{-1} J(x)^T \mu. \tag{20}$$

For $k = 2$, there are—after normalization—only two choices for μ with the property (17):

$$\mu^{(1)} = \begin{pmatrix}-1\\ 1\end{pmatrix}, \quad \text{and} \quad \mu^{(2)} = \begin{pmatrix}1\\ -1\end{pmatrix} \tag{21}$$

If all objectives are strictly convex, we can use μ to determine the orientation of ν_μ which we can see as follows: For ν_μ it is

$$\begin{aligned}(d_{\nu_\mu})_i &= (J(x)\nu_\mu)_i = -\nabla f_i(x)^T H_\alpha^{-1} J(x)^T\mu\\ &= -\sum_{j=1}^k \mu_j \nabla f_i(x)^T H_\alpha^{-1} \nabla f_j(x)\end{aligned} \tag{22}$$

Since all the f_i's are strictly convex it follows that also H_α^{-1} is positive definite. Using (12) we obtain for the first component

$$\begin{aligned}(d_{\nu_\mu})_1 &= -\mu_1 \nabla f_1(x)^T H_\alpha^{-1} \nabla f_1(x) - \mu_2 \nabla f_1(x)^T H_\alpha^{-1} \nabla f_2(x)\\ &= -\mu_1 \underbrace{\nabla f_1(x)^T H_\alpha^{-1} \nabla f_1(x)}_{>0} - \mu_2 \underbrace{\lambda}_{<0} \underbrace{\nabla f_1(x)^T H_\alpha^{-1} \nabla f_1(x)}_{>0}\end{aligned} \tag{23}$$

Evaluating the choices for μ in (21) we see that

$$(d_{\nu_\mu})_1 = \lambda_1 \mu_1, \tag{24}$$

where $\lambda_1 < 0$. Analog, we obtain

$$(d_{\nu_\mu})_2 = \lambda_2 \mu_2, \tag{25}$$

for a $\lambda_2 < 0$. Thus, μ determines the orientation of ν_μ measured by the orientation in objective space. For instance, for $\mu^{(1)} = (-1, 1)^T$ we obtain a 'right lower' movement along the Pareto front. Using (11) we see further that the movement in objective space is orthogonal to α and in the 'opposite' direction of μ.

We put the above consideration together to the following result.

Theorem 1. *Let $x \in \mathbb{R}^n$ be a Pareto point of a bi-objective problem and $\alpha \in \mathbb{R}^2$ its associated KKT multipliers. Further, let $\alpha_i \neq 0$, $i = 1, 2$, and $\nabla f_i(x) \neq 0$, $i = 1, 2$ and let all objectives be strictly convex. For μ as in (21) the vector*

$$\nu_\mu = -H_\alpha^{-1} J(x)^T \mu. \tag{26}$$

points along the linearized Pareto set, and

$$d_{\nu_\mu} = -\mu_1 \xi \begin{pmatrix} \alpha_2 \\ -\alpha_1 \end{pmatrix}, \tag{27}$$

where $\xi \in \mathbb{R}_+$, is its corresponding movement in objective space as in (9).

Figure 2 shows the directions ν_μ from (20) along the Pareto set of MOP (14).

4 Chosen Predictor Corrector Variants and Numerical Results

Based on the above discussion we will in the sequel consider the following four PC variants

1. **Classical MO Continuation**
   Here we follow the steps described in Section 2.2 applied on the map $\tilde{F}$ defined in (15) on the (x, α)-space. For the corrector we use the Levenberg-Marquardt algorithm.
2. **Newton Continuation**
   For the tangent we follow (20) and set for the predictor at a given solution x:

$$p(x) = -\left(\sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x)\right)^{-1} J(x)^T \mu, \tag{28}$$

   where we use μ to steer the search. The corrector is computed using the multi-objective Newton method proposed in [19].

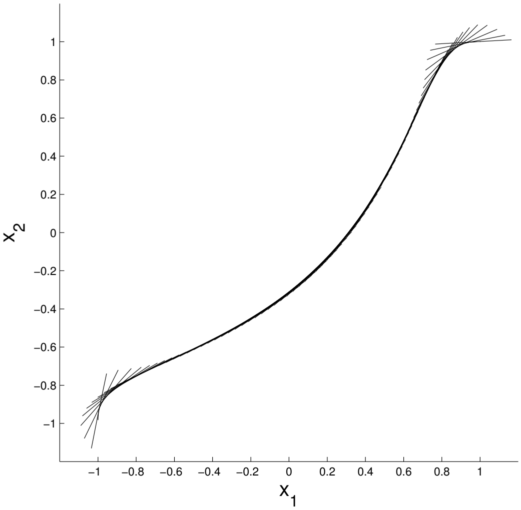


Fig. 2. Directions ν_μ from (20) along the Pareto set of MOP (14)

3. **Quasi-Newton Continuation**
   This is the 'Quasi-Newton version' of the above algorithm: For the tangent at x we choose
   $$p(x) = -\left(\sum_{i=1}^{k} \alpha_i B_i(x)\right)^{-1} J(x)^T \mu, \tag{29}$$
   where $B_i \approx \nabla^2 f_i(x)$ is computed via a BFGS update ([21]) and where μ is used to steer the search. For the corrector we take the Quasi-Newton Algorithm for MOPs proposed in [20].
4. **Steepest Descent Continuation**
   For the tangent space we choose one of the gradients according to the movement of the search
   $$p(x) = -\nabla f_i(x) \tag{30}$$
   and for the corrector we take the steepest descent method presented in [18].

For all methods we have taken the step size ([1])

$$h = \min_{i=1,2} \frac{d}{J(x)p_N(x)}, \tag{31}$$

where $d > 0$ and $p_N(x)$ denotes the normalized tangent vector of each function and have taken the predictor

$$p = x + hp_N(x). \tag{32}$$

Note that step size (31) is chosen such that

$$\|F(p) - F(x)\| \approx d \tag{33}$$

for a chosen value of d. For the Newton and Quasi-Newton methods we have chosen the strong Wolfe condition with cubic interpolation and for the Steepest Descent method we have taken the Armijo condition with cubic interpolation. In all cases we have chosen $5\sqrt{eps}$ as the tolerance for the stop condition.

First we consider the bi-parameter BOP ([23])

$$\begin{aligned} f_1(x) &= \frac{1}{2}(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} + x_1 - x_2) + \lambda \cdot e^{-(x_1-x_2)^2} \\ f_2(x) &= \frac{1}{2}(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} - x_1 + x_2) + \lambda \cdot e^{-(x_1-x_2)^2}, \end{aligned} \tag{34}$$

where $\lambda = 0.85$ and the domain was chosen to be $Q = [-8, 8]^2$. The Pareto set forms a line segment connecting the points $(-8, 8)^T$ and $(8, -8)^T$. Figure 3 shows the Pareto set and front as well as the numerical result obtained by the Quasi-Newton continuation. It has to be mentioned that in all cases considered here all the PC variants obtained the same approximation quality. This does, however, not hold for the computational efforts (refer to Table 1): First, it is apparent that the cost of the Newton continuation method is much less than the one from the classical continuation which results from the different number of corrector steps. The main reason for this difference may be that $\tilde{F}$ is defined in the compound (x, α) space while the Newton continuation—once the value of μ is set—does not require the weight space. A potential problem for the compound space is that its sets have different magnitudes. While the set (and thus its magnitude) in α-space is fixed, this does apparently not hold for the Pareto set/front which depends on the model. Assume the direction $\nu = (\nu_x, \nu_\alpha) \in \mathbb{R}^{n+k}$ is obtained via (4), then the predictor in the classical approach is given by

$$\begin{pmatrix} p_x \\ p_\alpha \end{pmatrix} = \begin{pmatrix} x_0 \\ \alpha_0 \end{pmatrix} + t \begin{pmatrix} \nu_x \\ \nu_\alpha \end{pmatrix}, \tag{35}$$

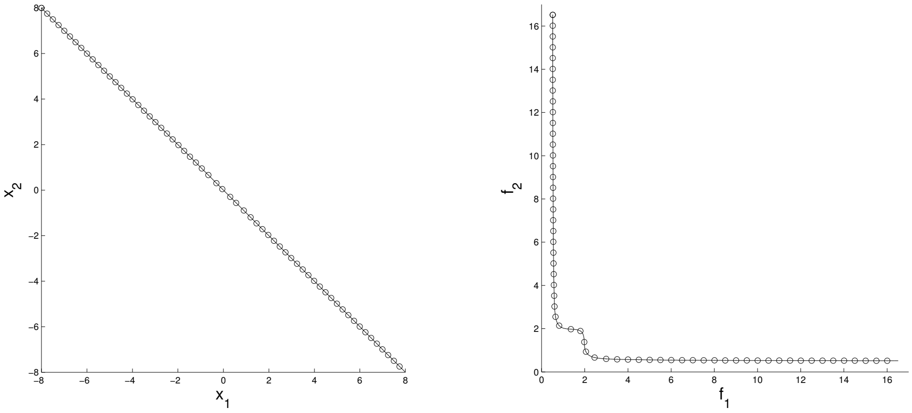
where t is chosen to satisfy (33) which certainly depends on the magnitude of the Pareto set/front. Thus, even if the Pareto set is linear, it is not necessarily given that $\tilde{F}(p_x, p_\alpha) = 0$ which may result in additional corrector steps.

On the other hand, the predictor chosen by the Newton method is a Pareto point in case the Pareto set is linear. In this case, only one corrector step was used in total. This was when the Pareto set was abandoned in the last step of the continuation. Since for general MOPs we can consider the Pareto set to be close to linear on a small enough stretch, we can hence expect less corrector steps for the Newton continuation method than for the classical variant, at least for sufficiently small predictor steps.

The second observation is that the computational effort is further decreased by the two methods that are Hessian free.

The above observations get confimed by the next 10-dimensional BOP ([24])

$$f_j(x) = \sum_{i=1}^{10} (x_i - a_i^j)^2, \quad j = 1, 2, \tag{36}$$



(a) Parameter Space

(b) Objective Space

Fig. 3. Numerical results of the Quasi-Newton continuation on MOP (34)

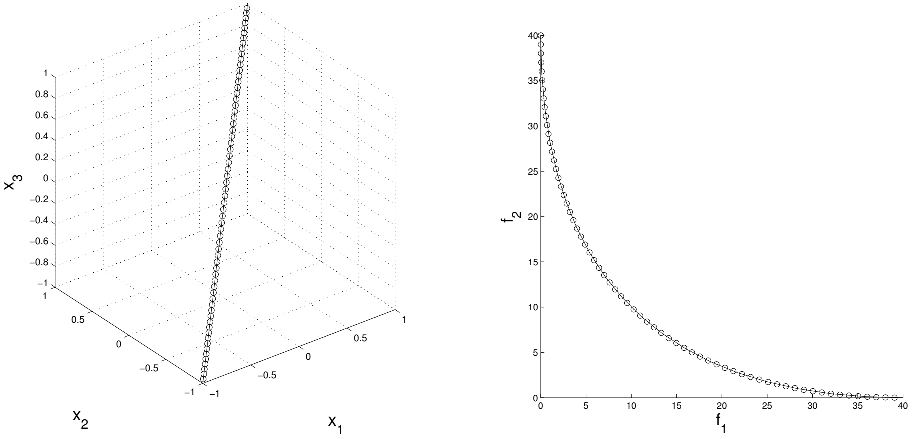
Table 1. Numerical results by the four PC variants on MOP (34)

	Classical	Newton	Quasi-Newton	Steepest Descent
No. of iterations	65	64	64	64
No. of solutions	65	63	63	63
Avg. corrector its.	2.353846	0.046875	0.046875	0
Avg. backtrack its.	-	0	0	0
Function evaluations	65	66	66	63
Jacobian evaluations	218	66	66	63
Hessian evaluations	434	131	-	-

Table 2. Numerical results by the four PC variants on MOP (36)

	Classical	Newton	Quasi-Newton	Steepest Descent
No. of iterations	65	65	66	65
No. of solutions	66	66	66	66
Avg. corrector its.	0.846154	0.015385	0.030303	0.015385
Avg. backtrack its.	-	0	0	0.015385
Function evaluations	66	67	69	68
Jacobian evaluations	307	67	69	67
Hessian evaluations	612	133	-	-

where $a^1 = (1, \ldots, 1)^T \in \mathbb{R}^{10}$ and $a^2 = -a^1$. The Pareto set is a line segment connecting a^1 and a^2 (i.e., linear). Figure 4 shows the numerical result of the Newton continuation and Table 2 shows the computational efforts of each method. Again, we observe that (a) the Newton continuation is cheaper than the classical continuation method due to the fewer number of corrector steps and that (b) the computational effort can be further significantly reduced when using one of the Hessian free methods.



(a) Parameter Space

(b) Objective Space

Fig. 4. Numerical results of the Quasi-Newton continuation on MOP (36)

Table 3. Numerical results by the four PC variants on MOP (37)

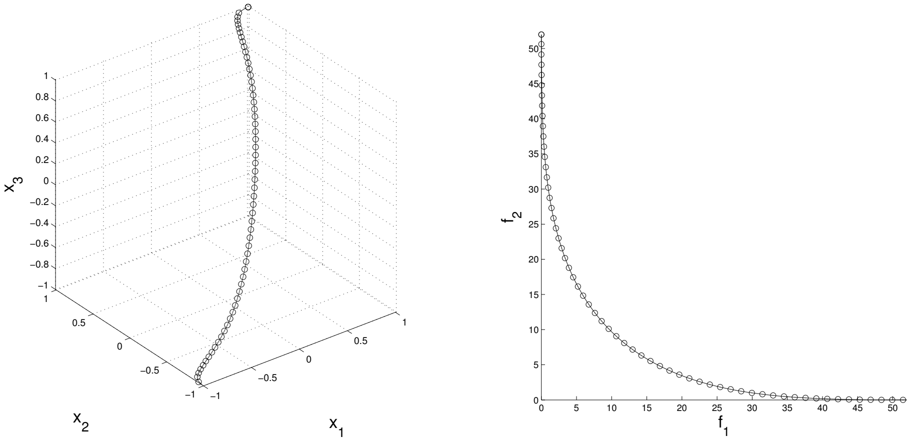
	Classical	Newton	Quasi-Newton	Steepest Descent
No. of iterations	60	60	61	68
No. of solutions	60	61	62	69
Avg. corrector its.	2.65	1.116667	1.524590	11
Avg. backtrack its.	-	0	0.004098	0.539988
Function evaluations	61	128	156	1080
Jacobian evaluations	220	128	156	817
Hessian evaluations	438	255	-	-

Next we consider the following problem ([25]):

$$f_j(x) = \sum_{\substack{i=1\\ i\neq j}}^{10} (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \quad j = 1, 2, \tag{37}$$

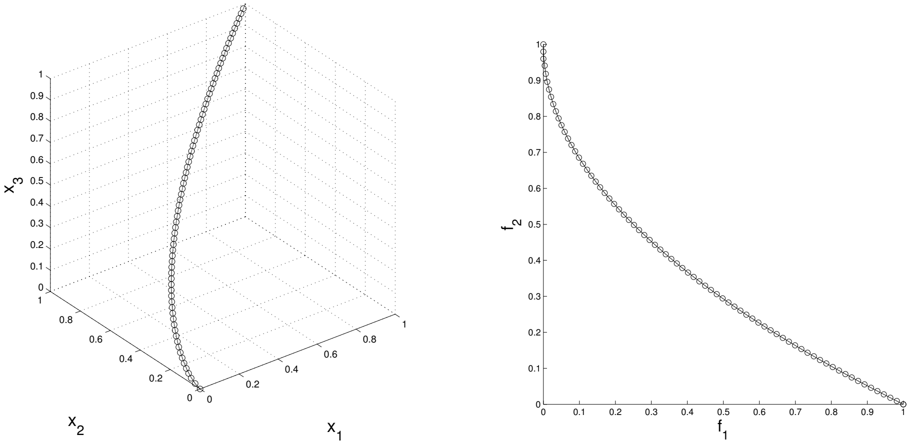
where a^1 and a^2 are as above. The Pareto set of this problem is a non-linear curve connecting the two minima a^1 and a^2. See Figure 5 and Table 3 for the result of the Newton continuation and the computational efforts, respectively. Though the Pareto set is non-linear we see again that the Newton continuation outperforms the classical PC method, and that the cost can be further reduced via using its Quasi-Newton counterpart. Since this model is more complicated than the other ones, the steepest descent method needs much more corrector steps to bring a predicted solution back to the solution set.

Finally, we consider problem F1 from [26] which confirms again the above observations made for a problem with non-linear Pareto set. See Figure 6 and Table 4 for more details.



(a) Parameter Space (b) Objective Space

Fig. 5. Numerical results of the Quasi-Newton continuation on MOP (37)



(a) Parameter Space (b) Objective Space

Fig. 6. Numerical results of the Quasi-Newton continuation on MOP F1

We stress that we intended to make a comparison to the PC-like method proposed in [17] that uses the gradient as predictor direction. Unfortunately, we were not able to produce results with a similar approximation quality since we e.g. got outcome sets of significantly larger or smaller magnitudes depending on the choice of some design parameters. The inclusion of this algorithm—as well as other related ones—into the comparison is subject of future studies.

Table 4. Numerical results by the four PC variants on MOP F1

	Classical	Newton	Quasi-Newton	Steepest Descent
No. of iterations	72	72	70	73
No. of solutions	71	71	69	72
Avg. corrector its.	3.069444	0.569444	0.9	4.808219
Avg. backtrack its.	-	0	0	0.925859
Function evaluations	71	112	132	737
Jacobian evaluations	292	112	132	423
Hessian evaluations	582	223	-	-

5 Conclusions and Future Work

In this paper, we have investigated two choices for the predictor direction within continuation methods for the numerical treatment of bi-objective optimization problems. The first one is the gradient direction at a given Pareto optimal solution x as proposed in [17]. By this, a movement along the linearized Pareto front at $F(x)$ is obtained, and we have shown that the gradient is the greedy choice for the movement in objective space. On the other hand, such predictors do not point along the Pareto *set* in certain cases leading to a potential loss in the overall performance. As second predictor we have proposed a variant of the classical choice used for the continuation of general implicitly defined manifolds. The new choice has the advantage that the weight space coming from the Karush-Kuhn-Tucker equations does not have to be considered in an augmented system.

Based on these observations we have tested three continuation methods and have compared them to the classical multi-objective continuation method on four unconstrained bi-objective problems. The numerical results indicate that significant savings in the computational effort compared to the classical approach can be achieved by the continuation method that includes the second predictor choice combined with Newton's method for MOPs ([19]), called 'Newton continuation' here for sake of a distinction. The best overall performance among the four methods is obtained by the Quasi-Newton variant of the Newton continuation.

Based on these results for unconstrained bi-objective problems the next steps are straightforward and include efficient constraint handling techniques as well as the application to problems with more than two objectives. Further, we intend to intensify the investigation in particular of the Quasi-Newton continuation method.

Acknowledgement. The first author acknowledges support from CONACyT through a scholarship to pursue undergraduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 128554.

References

[1] Hillermeier, C.: Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach. Birkhäuser (2001)

[2] Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization 8, 631–657 (1998)

[3] Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)

[4] Fliege, J.: Gap-free computation of Pareto-points by quadratic scalarizations. Mathematical Methods of Operations Research 59, 69–89 (2004)

[5] Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Heidelberg (2008) ISBN 978-3-540-79157-7

[6] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X

[7] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007) ISBN 978-0-387-33254-3

[8] Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. Journal of Optimization Theory and Applications 124, 113–155 (2005)

[9] Jahn, J.: Multiobjective search algorithm with subdivision technique. Computational Optimization and Applications 35(2), 161–175 (2006)

[10] Schütze, O., Vasile, M., Junge, O., Dellnitz, M., Izzo, D.: Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. Engineering Optimization 41(2), 155–181 (2009)

[11] Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., Sun, J.Q.: Simple cell mapping method for multi-objective optimal feedback control design. International Journal of Dynamics and Control 1(3), 231–238 (2013)

[12] Hernández, C., Sun, J.-Q., Schütze, O.: Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 171–188. Springer, Heidelberg (2013)

[13] Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer (1990)

[14] Recchioni, M.C.: A path following method for box-constrained multiobjective optimization with applications to goal programming problems. Mathematical Methods of Operations Research 58, 69–85 (2003)

[15] Schütze, O., Dell'Aere, A., Dellnitz, M.: On continuation methods for the numerical treatment of multi-objective optimization problems. In: Branke, J., Deb, K., Miettinen, K., Steuer, R.E. (eds.) Practical Approaches to Multi-Objective Optimization. Number 04461 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany (2005), http://drops.dagstuhl.de/opus/volltexte/2005/349

[16] Pereyra, V.: Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems. Math. Comput. Simul. 79(6), 1935–1947 (2009)

[17] Wang, H.: Zigzag search for continuous multiobjective optimization. INFORMS J. on Computing 25(4), 654–665 (2013)

[18] Drummond, L.M.G., Svaiter, B.F.: A steepest descent method for vector optimization. J. Comput. Appl. Math. 175, 395–414 (2005)

[19] Fliege, J., Drummond, L.M.G., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM J. on Optimization 20, 602–626 (2009)

[20] Povalej, Ž.: Quasi-Newtons method for multiobjective optimization. Journal of Computational and Applied Mathematics 255, 765–777 (2014)

[21] Nocedal, J., Wright, S.: Numerical Optimization. Series in Operations Research and Financial Engineering. Springer (2006)

[22] Pareto, V.: Manual of Political Economy. The MacMillan Press (1971 (original edition in French in 1927))

[23] Witting, K.: Numerical Algorithms for the Treatment of Parametric Multiobjective Optimization Problems and Applications. Dissertation, Universitt Paderborn (February 2012)

[24] Köppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)

[25] Schütze, O.: Set Oriented Methods for Global Optimization. PhD thesis, University of Paderborn (2004),
http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/

[26] Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. IEEE Transactions on Evolutionary Computation 13(2), 284–302 (2009)

Part VI

Genetic Programming

Application of Genetic Programming Models Incorporated in Optimization Models for Contaminated Groundwater Systems Management

Bithin Datta[1,2,*], Om Prakash[1,2], and Janardhanan Sreekanth[3]

[1] Discipline of Civil and Environmental Engineering, School of Engineering
and Physical Sciences, James Cook University, Townsville QLD 4811, Australia
Bithin.datta@jcu.edu.au
[2] CRC for Contamination Assessment and Remediation of the Environment,
Mawson Lakes SA 5095, Australia
[3] CSIRO Land and Water Division, Dutton Park, Queensland 4102, Australia

Abstract. Two different applications of Genetic Programming (GP) for solving large scale groundwater management problems are presented here. Efficient groundwater contamination management needs solution of large sale simulation models as well as solution of complex optimal decision models. Often the best approach is to use linked simulation optimization models. However, the integration of optimization algorithm with large scale simulation of the physical processes, which require very large number of iterations, impose enormous computational burden. Often typical solutions need weeks of computer time. Suitably trained GP based surrogate models approximating the physical processes can improve the computational efficiency enormously, also ensuring reasonably accurate solutions. Also, the impact factors obtained from the GP models can help in the design of monitoring networks under uncertainties. Applications of GP for obtaining impact factors implicitly based on a surrogate GP model, showing the importance of a chosen monitoring location relative to a potential contaminant source is also presented. The first application utilizes GP models based impact factors for optimal design of monitoring networks for efficient identification of unknown contaminant sources. The second application utilizes GP based ensemble surrogate models within a linked simulation optimization model for optimal management of saltwater intrusion in coastal aquifers.

Keywords: Optimal Monitoring Network, Groundwater Pollution, Genetic Programming, Multi-Objective Optimization, Pollution Source Identification, Simulated Annealing, Impact Factors, Ensemble Surrogates.

1 Introduction

Two significant applications of Genetic Programming (GP) towards development of optimal solutions to Water Resources Management problems are presented here. Two applications are presented: using the impact factors [12] obtained from trained GP

[*] Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_13, © Springer International Publishing Switzerland 2014

models as implicit surrogate models for optimal design of monitoring networks dedicated to identifying unknown groundwater contamination sources; and using trained ensemble GP models to develop surrogate models within a linked simulation optimization model for management of coastal aquifers. The first application essentially uses trained GP models to compute the impact of any potential monitoring location in relation to a possible contaminant source. The second application addresses a major problem faced by researchers in the large scale water resources management area i.e., the necessity of simulating the physical processes involved, within the optimization model. Classical nonlinear constrained optimization techniques are not ideally suited to the linking of the physical processes descriptive models, e.g., numerical simulation models; and the decision making prescriptive optimization models [6]. Evolutionary optimization techniques like Genetic Algorithm (GA) and Simulated Annealing (SA) or Adaptive Simulated Annealing (ASA) provided a much more simple way of developing linked simulation optimization models in which the linking process is greatly simplified due to the optimal solution search process in the evolutionary algorithms. Use of GP models along with the evolutionary optimization algorithms can make it possible to drastically improve computational efficiency and ensure computational feasibility in solution of large scale water resources management problems.

Using GA or SA greatly simplifies the process of linking the simulation and optimization models. However, the nature of the repetitive search process makes the solution of the linked optimization model and the numerical simulation model computationally intensive and time consuming [11]. One way to overcome this requirement of very large computational time is the use of surrogate or meta models trained and implemented as approximations of the numerical simulation models. These surrogate models can be linked with the optimization model. This results in enormous savings in computational time while maintaining the solution inaccuracies within acceptable limits. In the past, trained and tested Artificial Neural Network (ANN) models have been proposed as surrogate models to approximate the flow and transport process in water resources systems. More recently, trained Genetic Programming (GP) based surrogate models have been proposed for computational efficiency [17]. In addition, ensemble surrogate GP models s can be effectively utilized as efficient alternatives to the direct use of numerical simulation models. Ensemble surrogate GP model also facilitates the estimation of the reliability of the prescriptive solutions obtained as solution of the optimization model under various uncertainties.

GP based, trained surrogate or meta models can describe the approximate impacts of particular inputs on the resulting output patterns. These GP impact factors provide an excellent means for optimal design of monitoring network based on budgetary constraints, and reducing the redundancy of monitoring locations in the optimal design [7].

Two recently proposed application of GP models are presented here. These are: (i) development of optimal and efficient monitoring network designs for detection of pollutant plumes and identification of unknown pollution source locations in contaminated groundwater systems, and (ii) use of ensemble GP based surrogate models within a linked simulation optimization models for management and control of saltwater intrusion in coastal groundwater aquifers. These two illustrative applications of GP models as well as evolutionary algorithms based linked simulation optimization models are presented and discussed here. Linked simulation optimization models (LSOM) have been previously utilized for solving related problems such as, simultaneous estimation of unknown pollution sources and estimation of unknown aquifer parameters. Successful application of optimization and ANN techniques to solve this problem is demonstrated in [16]. Some of the earlier attempts to use ANN based surrogate models within a linked simulation optimization framework for developing optimal management strategies for control of saltwater intrusion in coastal aquifers has been reported in [3]and [17]. The GP based approach discussed here represents improvement in the scope and efficiency of earlier ANN based approaches.

1.1 The Physical Problems Addressed

Two challenging problems in the management of large scale contaminated groundwater aquifers are: (i) design of an optimal monitoring network for efficient collection of contamination measurements under various uncertainties e.g., parameter uncertainty, modelling uncertainty, and measurement errors, and (ii) designing an optimal regional management strategy for sustainable beneficial use of coastal aquifers incorporating the complex nonlinear density dependent flow and transport processes in coastal aquifers. The use of GP models in these two different applications to the management of contaminated aquifers is described here. In this study, Linear Genetic Programming (LGP) algorithm [10] was utilized. Some of the GP parameter values for training validation and testing are same as the default values given in [10].

2 Monitoring Network Design for Efficient Identification of Aquifer Contamination Sources

A number of linked simulation optimization models (LSOMs) have been proposed using different optimization algorithms for improving the source identification results as reported in [2]. Very limited amount of work has been reported for improving the accuracy of source identification by using pollutant concentration measurements from an optimally designed monitoring network in a linked simulation-optimization approach.

In real world problems the number of monitoring locations is governed by budgetary constraints. Therefore it is important that the monitoring locations are chosen such that the concentration measurements at these locations when used in linked simulation-optimization method improve the accuracy of source identification results. The GP impact factor based methodology has been developed for improving the accuracy of source identification results using linked simulation-optimization technique. This methodology uses trained GP models to calculate the impact factor of the sources on the candidate monitoring locations. The Pareto-optimal solutions obtained from the two objective models are utilized to design a Pareto-optimal monitoring network. Simulated Annealing (SA) based linked simulation-optimization is used for source identification purpose. Fig. 1 shows a typical illustrative contaminated aquifer study area, the source locations, contaminant plumes, and the identified source fluxes for each stress period considered.

2.1 Methodology for Optimal Monitoring Network Design

In the first step, GP models are trained utilizing a large set of data pattern comprising of source flux history for all the potential sources as input and corresponding aquifer response at all potential monitoring locations as the output. Based on the fitness value, a specific number of best GP models are chosen for calculating the impact factor of a potential source on a monitoring location. The impact factor is calculated for all monitoring locations at each monitoring time step. A multi–objective optimization formulation is applied to select the optimal set of monitoring location for source identification using linked simulation-optimization technique.

In the second step, a SA based linked simulation-optimization model is solved in which the difference between the simulated and measured pollutant concentrations at these optimally chosen monitoring locations are minimized. The same problem is also solved for different arbitrary monitoring locations and the results are compared.

Performance of the proposed source identification methodology is evaluated by solving an illustrative problem.

Genetic Programming and Impact Factor

Specified number of best individual GP models is used for computing the impact factor. The impact factor is described as a measure of how much an input variable accounts for the output result i.e., a factor by which the result would differ if the variable was removed. This essentially implies that, if by removing a variable from the mathematical function (GP model) there is a large change in the output, then the removed variable has a high impact on the output and hence the impact factor of that variable will be high. The impact factor value is then used as design criteria in a multi-objective optimization formulation for designing an optimal monitoring network.

Multi-Objective Optimization Model or Monitoring Network Design Model

The multi-objective optimization model that finds monitoring well locations with the following objectives (1) finding well locations with maximum normalized impact from all the potential sources and (2) finding well locations with maximum normalized relative impact from an individual potential source over a chosen observation period. Objective 1) reduces the possibility of missing an actual source as it chooses locations where overlapping of plumes due to potential sources is maximum. This also reduces the likelihood of choosing monitoring locations where the impact of potential sources are small. Objective 1 is in conflict with objective 2 of finding well locations with maximum normalized relative impact from an individual potential source. The formulation for the multi-objective optimization for monitoring network design given in Eq. (1) through Eq. (9).

$$IF_{iob}^{S} = \sum_{t=1}^{nt} (F_{iob}^{St}) \tag{1}$$

IF_{iob}^{S} is the impact factor of source *S* on monitoring well location *iob*; F_{iob}^{St} is the impact factor of source *S* on monitoring well location *iob* at stress period *t*, $SumIF_{iob}^{S}$ is the sum of the impact factors of a potential source *S* at any given monitoring location *iob* for *nk* sampling steps

$$SumIF_{iob}^{norm} = \sum_{S=1}^{nS} \frac{SumIF_{iob}^{S}}{\frac{1}{nob}\sum_{iob=1}^{nob} SumIF_{iob}^{S}} \tag{2}$$

$SumIF_{iob}^{norm}$ is the normalised sum of impact factor at any monitoring location *iob* due to all the potential sources *nS* for all *nk* monitoring time steps, *nt* is the total number of stress periods; *nk* is the total number of monitoring time steps;

$\frac{1}{nob}\sum_{iob=1}^{nob} SumIF_{iob}^{S}$ is the average impact factor due to a source *S* at all monitoring well locations *nob*

$$^{\mathrm{Re}l}SumIF_{iob} = Max\{SumIF_{iob}^{S}\} - ((\sum_{S=1}^{nS}(SumIF_{iob}^{S})) - Max\{SumIF_{iob}^{S}\}) \tag{3}$$

$^{\mathrm{Re}l}SumIF_{iob}$ is the relative impact factor due to all the sources *nS* at a given monitoring well location *iob*

$$^{\mathrm{Re}l}SumIF_{iob}^{norm} = \frac{^{\mathrm{Re}l}SumIF_{iob}}{\frac{1}{nS}\sum_{S=1}^{nS} SumIF_{iob}^{S}} \tag{4}$$

$^{\mathrm{Re}l}SumIF_{iob}^{norm}$ is the normalized relative impact factor at monitoring well location *iob* for all potential sources;

$\frac{1}{nS}\sum_{S=1}^{nS} SumIF_{iob}^{S}$ is the average impact factor at monitoring well location *iob* for all potential sources.

The two objectives *F1* and *F2* of the multi-objective optimization model for optimal monitoring network design for accurate identification of unknown pollution sources is defined by Eq. (5) and Eq. (6) respectively.

$$Maximize F\,1 = \sum_{iob=1}^{nob} SumIF_{iob}^{norm} f_{iob} \tag{5}$$

$$Maximize F\,2 = \sum_{iob=1}^{nob} {}^{\mathrm{Re}\,l} SumIF_{iob}^{norm} f_{iob} \tag{6}$$

Subject to:

$$\sum_{iob=1}^{nob} f_{iob} \leq \alpha \tag{7}$$

α is integer constant representing the maximum number of wells that can be chosen; f_{iob} represent the binary decision variable to select a monitoring well location $f_{iob} \equiv \{0,1\}$ such that when f_{iob} value equal to 1 representing monitoring well to be selected at location *iob*, and zero otherwise.

The two objective multi-objective optimization model is solved using the constrained method. All the solutions lying on the Pareto-optimal front corresponds to a different monitoring network.

Linked Simulation-Optimization Model for Source Identification

Groundwater flow (MODFLOW) and solute transport (MT3DMS) simulation models are used to simulate the physical process within the optimization model. Simulated Annealing (SA) is used as an optimization algorithm to solve the optimization problem. The optimization model minimizes the difference between the observed contaminant concentration and the simulated concentrations resulting from the optimal source (decision variables) fluxes subject to the linked simulation constraints.

$$Minimize F = \sum_{k=1}^{nk} \sum_{iob=1}^{nob} ABS(cest_{iob}^{k} - cobs_{iob}^{k}) \tag{8}$$

$$cest_{iob}^{k} = f(q_s, C_s) \quad f(q_s, C_s, cest_{iob}^{k}) \;\forall\; iob, k \tag{9}$$

$q_s C_s$ is the pollutant source fluxes; q_s is the volumetric flux; C_s is the concentration of the sources or sinks; ABS is the absolute difference; $cest_{iob}^{k}$ is the simulated concentration; nk is the total number of monitoring time steps; nob is the total number of observation wells; $cobs_{iob}^{k}$ is the observed concentration.

2.2 Performance Evaluation

The performance of the developed methodology was evaluated for the study area shown in Fig. 1. Three sources with three stress periods of 500 days each were considered. The pollutant flux from each of the sources is assumed to be constant over a stress period. Four temporal concentration measurements at each potential location (starting t = 1600 days) are obtained after every 200 days. Details are available in [15].

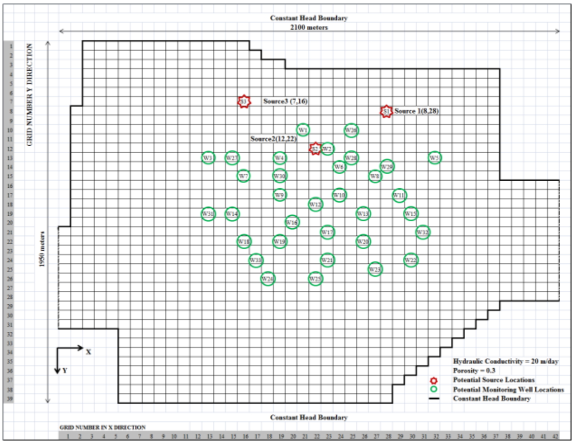


Fig. 1. Plan view of the Illustrative Study Area

Computation of Genetic Programming Impact Factors

The impact factor is calculated for 3 contaminant sources and 25 (W1 to W25 as shown in Fig. 1) potential monitoring well locations. The input data consists of flux values for each source at every stress period. The corresponding output data consists of the resulting pollutant concentration measurement due to these source fluxes at all the 25 potential monitoring well at different specified times. 3000 data patterns are used of which 50% is used for training, 40% for validation, and 10% for testing the GP models. Based on the fitness value, top 30 GP models are used for computing the impact factor. The impact factor for all the potential monitoring locations at every sampling time step is calculated likewise which is then used to calculate the normalized relative impact factor ${}^{\mathrm{Re}l}SumIF_{iob}^{norm}$ and normalised sum of impact factor due to all the potential sources $SumIF_{iob}^{norm}$.

Source Identification using Data from Pareto-Optimal Monitoring Networks and Arbitrary Monitoring Network

To evaluate the performance of the methodology, observed concentration measurements are generated synthetically. These simulated measurements are then perturbed with random error term (maximum deviation of 10 percent of the measured concentration data) to incorporate realistic measurement errors (Eq. 10). The linked simulation-optimization model is solved using measurements from 12 Pareto-optimal monitoring networks (MN1to MN12) and 10 arbitrary monitoring networks (ARMN1to ARMN10). The source identification model is solved with error free data and perturbed erroneous data.

$$ {}^{Pert}cobs_{iob}^{k} = cobs_{iob}^{k} + err \; ; \quad err = \mu per \times rand \tag{10} $$

$^{Pert}cobs_{iob}^{k}$ is the perturbed concentration value; $cobs_{iob}^{k}$ is the numerically simulated concentration value; err is the error term; $\mu per$is the specified maximum deviation expressed as a fraction < 1; $rand$ is a random fraction between -1 and +1.

2.3 Results and Discussion

The Pareto-optimal solution for the two-objective (*F1*and *F2*) optimal monitoring network design model is shown in Fig. 2. The non-inferior solutions show the conflicting nature of the two objective functions and their trade-off. The results of source flux identification solution results obtained by using linked simulation-optimization model were compared for all the 12 Pareto-optimal monitoring networks (MN1 to MN12), obtained as solutions, using error free and perturbed error data and the solution results. The source estimations were reasonably accurate.

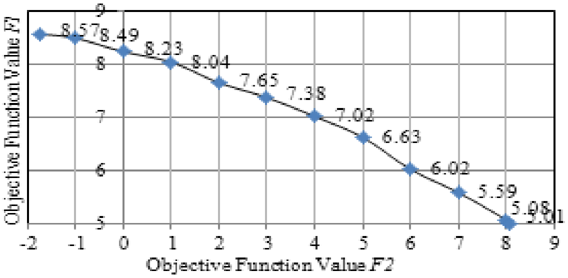


Fig. 2. Pareto Optimal Solution Front

To illustrate the advantage of using a designed optimal monitoring network, e.g., MN5, the source identification model was solved for 10 arbitrary networks (ARMN1 to ARMN10) with both error free data and erroneous data. The estimated flux values using the arbitrary networks is averaged and compared with the actual flux values and estimated flux value from monitoring network 5 (MN5), both for error free data and erroneous data (Fig. 3a and 3b respectively). It is seen that the estimated flux using monitoring network 5 (MN5), are closer to the actual flux values as compared to the flux estimated using the arbitrary networks (AVG-AR).

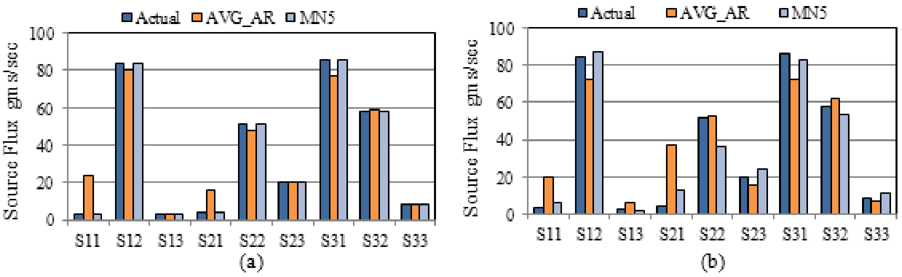


Fig. 3. Comparison of Identification Results using Optimal Network and Arbitrary Networks: Error free measurement data and erroneous data [15]

Therefore, the designed monitoring networks are reasonably accurate in estimating the unknown source fluxes. An application of GP models in management of saltwater intrusion in coastal aquifers is discussed in the following sections.

3 Surrogate-Based Coupled Simulation-Optimisation Framework for Coastal Aquifer Management

A number of studies have used density-dependent flow and transport models for developing coastal aquifer management models [4, 5]. [5] used an embedding approach to develop the management model. [9] used the linked density-dependent flow and transport simulation model FEMWATER for developing a multi-objective management model. [1] integrated the genetic algorithm optimisation technique with a three-dimensional density-dependent flow and transport simulation model to develop management scenarios to control saltwater intrusion. In order to increase the computational feasibility of the methodology, and address this computation time constraint which can often exceed few weeks for a typical study area, the surrogate model approach is adopted to approximate the flow and transport processes using trained GP models.

3.1 Genetic Programming as a Surrogate Modelling Tool

A few studies in the broad area of hydrology and water resources have used GP models [14]. From the limited number of studies in the broad area of hydrology and water resources, GP seems to be a simple and efficient tool for functional approximation. [17, 18, 19] developed GP based surrogate models to substitute three-dimensional density-dependent flow and transport simulation models for simulating pumping induced sa1twater intrusion processes within an optimisation framework.

One of the main objectives of this study was to develop computationally feasible simulation-optimisation methodologies for prescribing optimal pumping solutions for coastal aquifer management to control saltwater intrusion. Directly linking flow and transport simulation models to optimisation algorithms has been reported to incur a huge computational burden. [9] reported a run-time of 30 days for solving an illustrative coastal aquifer management problem based on simulation-optimisation where the numerical simulation model was linked directly to a multi-objective genetic algorithm for optimisation. GP based surrogate models are used here for approximating 3D density-dependent coupled flow and transport models within a multi-objective genetic algorithm to develop multi-objective optimal pumping strategies for coastal aquifer management. The three-dimensional density-dependent flow and transport simulation model FEMWATER [13] was chosen to simulate the coupled flow and transport process in the coastal aquifer system.

Coastal Aquifer Management Model

The coastal aquifer management essentially has two components. The first component is a surrogate model for predicting the salinity levels in the specified monitoring locations, as a result of the groundwater extraction from the aquifer. The second component is an optimisation algorithm based model to evolve optimal management strategies satisfying the imposed managerial constraints and other system constraints.

Optimal Pumping Strategies for Coastal Aquifers

The methodology addresses uncertainty in the values of hydraulic conductivity and groundwater recharge. The ensemble surrogate modelling approach together with multiple-realisation optimisation is utilized for stochastic optimisation under parameter uncertainty. Multi-objective optimal management considering beneficial and barrier well pumping is considered. The mathematical formulation of the problem considering parameter uncertainty is given as follows:

$$Maximize \sum_{p \in PROD} \sum_{t \in T} Q_t^p \tag{11}$$

$$Minimize \sum_{b \in BAR} \sum_{t \in T} q_t^b \tag{12}$$

$$c_i = f_i(Q_t^p, q_t^b, \boldsymbol{\theta}) \tag{13}$$

$$c_i \leq c_i^{\max} \tag{14}$$

$$Q_{\min} \leq Q_t^p \leq Q_{\max} \tag{15}$$

$$q_{\min} \leq q_t^b \leq q_{\max} \tag{16}$$

where, Q_t^p is the pumping from the p^{th} beneficial pumping well for the t^{th} time period, q_t^b is the pumping from the b^{th} barrier well for the t^{th} time period and c_i is the salinity at the monitoring location *i* at the end of the management time frame considered in the optimisation model, resulting from the pumping. PROD and BAR designate respectively, the set of all production wells and barrier wells in the well field. c_i is a function of Q_t^p and q_t^b and also the numerical model parameter set $\boldsymbol{\theta}$. The function f_i represents the numerical simulation model. When the parameter set $\boldsymbol{\theta}$ is considered as stochastic, solving this optimisation model would imply testing each solution comprising of a set of pumping values against multiple realisations of the uncertain parameter set $\boldsymbol{\theta}$. $Q_{\min}$ and $Q_{\max}$ are respectively the lower and upper limits on the production well pumping, and $q_{\min}$ and $q_{\max}$ are the corresponding values for the barrier well pumping.

Due to the computational difficulties in implementing this optimisation scheme, the ensemble surrogate model is used as an approximate substitute of the simulation model f_i within the optimisation model. The ensemble surrogate model based multiple realisation optimisation implements the reliability concept in the following manner.

$$c_i^r \approx \zeta_i^r (Q_t^p, q_t^b, U) \tag{17}$$

$$U = \psi(\boldsymbol{\theta}, \boldsymbol{\omega}) \tag{18}$$

$$\beta = {}^{r}\!\big/_{R} \tag{19}$$

$$c_i^r \leq c_i^{\max} \quad \forall r \;\; \text{such that} \; \sum r \geq R\beta \tag{20}$$

The concentration c_i is approximated using c_i^r, which are *r* different values of concentration at the i^{th} monitoring location, obtained from different surrogate models in the ensemble. The functional relationship between the pumping and the resulting salinity level is approximated by *r* realisations of the salinity obtained from different surrogate models given by ζ_i^r for each location *i*. The realisations c_i^r are different from each other because of the uncertainty *U* in the surrogate models, which is a function ψ of both numerical model parameters, $\boldsymbol{\theta}$ and the surrogate model structure and parameters, $\boldsymbol{\omega}$. Reliability can be defined as the ratio of number of realisations *r* which satisfies the constraint on the limit of concentration $c_i^{\max}$ to *R,* the total number of realisations of salinity obtained from the ensemble prediction models. Constraint (20) ensures that all realisations c_i^r which belongs to a set of realisations with a size of at least $r = R\beta$ should satisfy the limit on the concentration given by $c_i^{\max}$. Thus, the Pareto-optimal front for the multi-objective management problem is derived for a specific reliability level β, which is chosen by the manager depending on how reliable the solutions need to be.

Ensemble Surrogate Models

An ensemble of surrogates are developed using GP [20]. Each surrogate in the ensemble is trained and tested using a bootstrap sample set generated from the original training and testing data set. When these bootstrap samples, which contain repeated samples of pumping-salinity patterns, are used in the training, different weightings of patterns occur in the objective function used in the GP to develop the surrogate. Since the pumping-salinity patterns in the data pool correspond to different combination of uncertain parameters, it results in the development of surrogates which are differently capable of making predictions in different regions of the decision-parameter space.

All GP models used a population size of 500. The mutation and crossover probabilities used were respectively 0.95 and 0.5. The arithmetic operations - addition, subtraction, multiplication, division and the operators comparison data transfer were used as the functional set. The size of the terminal set was limited to 30 to prevent over fitting. The methodology of developing ensemble surrogates is illustrated in Fig. 4. With sufficient representation of the entire parameter and decision space in the original data set and sufficient number of surrogates in the ensemble, the ensemble surrogate modelling approach can achieve sufficiently accurate approximation of the saltwater intrusion prediction at the selected monitoring locations.

Coupled Simulation-Optimization

The surrogates developed using GP are coupled to the multi-objective genetic algorithm NSGA-II [8]. The methodology incorporating the parameter uncertainty is schematically represented in Fig. 5.

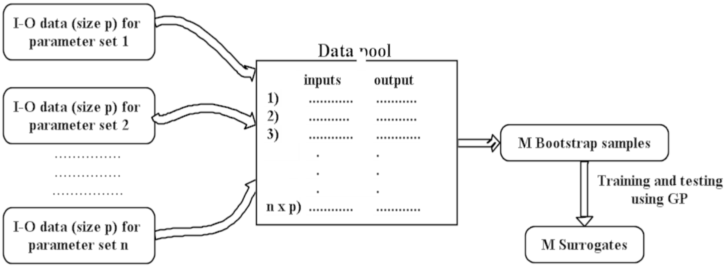


Fig. 4. Schematic representation of ensemble surrogate model used [20]

In the coupled simulation optimisation methodology all the surrogates are coupled with the multi-objective genetic algorithm NSGA-II in such a way that during each iteration of the optimisation search, all the surrogates are called upon for predicting the approximate value of the concentration. Thus, there are as many predictions of the concentration at a location, as the number of surrogates in the ensemble. Thus, multiple realisations of the concentration value are obtained from the ensemble of surrogates. The optimisation algorithm searches for the optimal pumping strategy which limits the concentration at the monitoring locations to prescribed values. In this multiple realisation approach, reliability is implemented as the percentage of the surrogates in the entire ensemble with concentration predictions that do not violate the imposed constraints of maximum concentration levels.

3.2 Performance Evaluation

The performance of the methodology is evaluated by applying it to a well field in a coastal aquifer system in the Lower Burdekin in Queensland, Australia [20]. Development of the conceptual model for simulating density-dependent flow and transport for the well field study area is described in the following section.

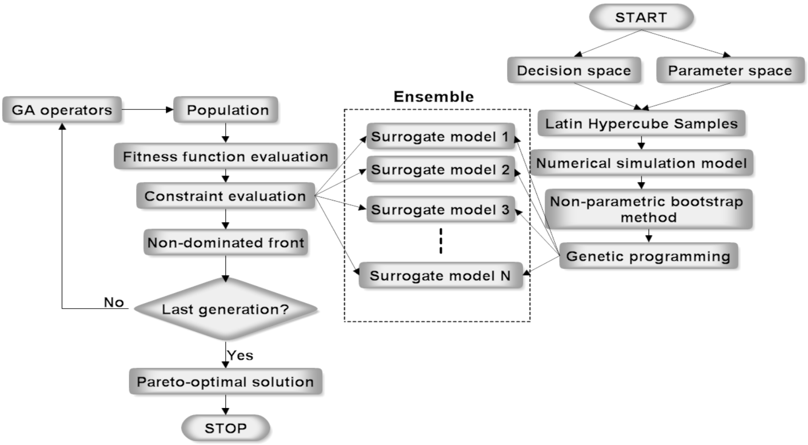


Fig. 5. Schematic representation of the ensemble based simulation-optimisation

The Aquifer Study Area

The Burdekin is a deltaic region in North Queensland extending over 850 km^2 in area and the primary land use is agriculture with extensive areas falling under sugarcane cultivation. About 40000 hectares of area is irrigated for cultivating sugarcane. For more than 120 years, the sugarcane industry has been active and groundwater has been extensively used for irrigation. In this work, Rita Island, in the Burdekin Delta, with an area of 60 km^2 was chosen for the performance evaluation study (Fig. 6). The area is essentially flat and is bounded by distributaries of the Burdekin River on two sides. On the eastern boundary of the study area is the Coral Sea. Part of the area is irrigated for sugarcane cultivation and the rest is an uncultivated area towards the sea.

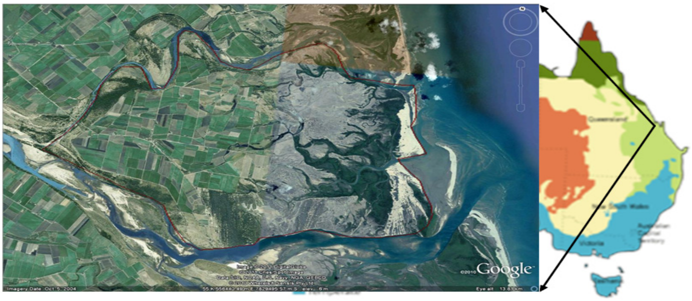


Fig. 6. Rita Island study area (Google Earth Pro, 2010)

Regional Scale Simulation Model

A regional scale groundwater flow and transport model was developed for the specific study area using the FEMWATER model. The study area with the boundaries and well locations are shown in Fig. 7.

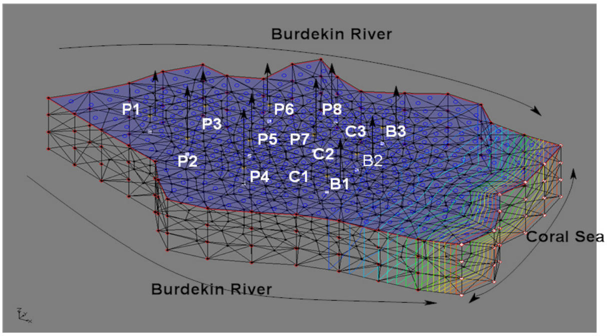


Fig. 7. Three-D view of the well field with well locations, control points and boundaries

Modelling Pumping-Induced Saltwater Intrusion

Altogether, 8 pumping well (irrigation) locations and 3 barrier well (to control hydraulic gradient) locations were considered within the study area. Uniform extraction rates are assumed at each location for a time-step of one year.

The aquifer responses were monitored at three nodes, at a depth of 19.8 metres, in between the pumping and barrier wells. The simulated concentration values at these nodes are named C1, C2 and C3 respectively as indicated in Fig. 8. The pumping management model prescribes permissible maximum concentration levels at these locations as 0.5, 0.6 and 0.6 kg/m^3 respectively. Corresponding to each combination of the uncertain parameters, 10 different pumping patterns were used to simulate the aquifer responses. Thus, 250 patterns of pumping were generated using Latin Hypercube Sampling and the FEMWATER model was used to simulate the aquifer processes to compute the concentration levels at the monitoring locations. Multiple training and testing sets were generated by using bootstrap sampling from the original data set of pumping-salinity patterns. For this, the original data set was split into two sets and bootstrap sampling was performed on each half to obtain separate training and testing sample sets. This ensures that training and testing of the surrogates are performed on mutually exclusive data sets.

3.3 Results and Discussion

Surrogate models were developed to predict the salinity concentration resulting from pumping at specified monitoring locations. Forty such surrogate models were developed for the prediction of salinities at each location C1, C2 and C3. Surrogate models were trained and tested using pumping-concentration data sets which were obtained from the simulation model models using different realisations of the uncertain parameters.

Root mean square error (RMSE) was calculated as an index to evaluate the predictive capability of the surrogate models in the ensemble. Coefficient of variation (CoV) of the RMSEs of the surrogate models in the ensemble was calculated as a measure of the uncertainty of the ensemble. The CoVs of the RMSEs in the prediction of C1 and C2 for increasing ensemble sizes are plotted in Fig. 8.

It is observed from Fig. 8 that, for predicting the salinity C1, there is a steady decrease in the value of the CoV of the RMSE with the increase in the number of surrogate models in the ensemble. The CoV is smallest when 37–40 surrogate models are present in the ensemble and it is observed that the corresponding slope of the CoV curve becomes close to zero indicating no further decrease. Therefore, the optimum number of models in the ensemble for predicting C1 was fixed as 40. Similar results were obtained for C2 and C3.

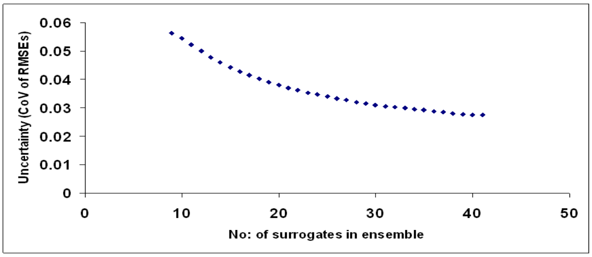


Fig. 8. Ensemble surrogate model uncertainty for salinity C1

Optimal Solutions with Different Reliabilities

An optimal solution to the pumping optimisation problem considering two conflicting objectives is a Pareto-optimal front of solutions which defines a trade-off between the two objectives. The Pareto-optimal front obtained, when the constraints imposed by all the 40 surrogate models in the ensemble are satisfied in the optimisation, is considered as the solution corresponding to a reliability level of 0.99. Similarly, optimal solutions with reliabilities 0.8 and 0.6 satisfy 32 and 24 out of the total 40 models, respectively. The Pareto-optimal set of solutions corresponding to these different reliability levels is shown in Fig. 10. This figure also shows the Pareto-optimal front corresponding to the single surrogate model based optimisation. This front appears to deliver better optimal solutions. However, when the corresponding pumping values are input in the numerical simulation model, it was observed that the all the solutions lying on the front violate some of the constraints. It was observed that this front

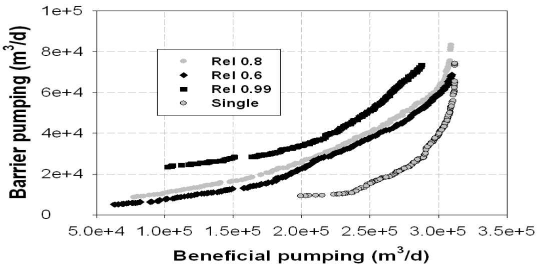


Fig. 9. Pareto-optimal fronts for different reliabilities and single surrogate modelling approach

actually represents solutions in the infeasible domain. This is primarily because of the inaccuracy in predictions by the single surrogate model. However, it was found that the RMSE values for the single surrogate models for C1, C2 and C3 were similar to that of the surrogate models in the ensemble.

4 Conclusions

A methodology using GP based impact factors, and SA based linked simulation optimization model has been developed for designing an optimal monitoring network for accurate source flux identification. Evaluation results show that contaminant concentrations measurements from such a designed monitoring network when used for source flux identification, can improve the accuracy of the source identification results. This methodology can increase the accuracy of source identification with concentration measurement data from limited number of monitoring wells in a designed monitoring network. In order to enhance the computation efficiency and ensure computational feasibility, trained GP models were developed as surrogate for the physical processes in a coastal aquifer system. Ensemble surrogate models addressed the uncertainties in modelling the system and can provide a means for estimating the reliability of the optimal management strategies obtained as solution for management of saltwater intrusion in coastal aquifers. This will lead to the development of robust optimal management strategies.

Acknowledgement. Authors would like to thank CRC-CARE, Australia and James Cook University, Australia for funding and support for this research.

References

1. Abd-Elhamid, H.F., Javadi, A.A.: A Cost-Effective Method to Control Seawater Intrusion in Coastal Aquifers. In: Water Resources Management, vol. 25, pp. 2755–2780 (2011)
2. Amirabdollahian, M., Datta, B.: Identification of contaminant source characteristics and monitoring network design in groundwater aquifers: an overview. J. of Environmental Protection 4(5A), 26–41 (2013)
3. Bhattacharjya, R.K., Datta, B.: ANN-GA-Based Model for Multiple Objective Management of Coastal Aquifers. Journal of Water Resources Planning and Management 135, 314–322 (2009)
4. Das, A., Datta, B.: 'Development of management models for sustainable use of coastal aquifers'. Journal of Irrigation and Drainage Engineering 125, 112–121 (1999a)
5. Das, A., Datta, B.: Development of multiobjective management models for coastal aquifers. Journal of Water Resources Planning and Management 125, 76–87 (1999b)
6. Datta, B., Chakrabarty, D., Dhar, A.: Simultaneous identification of unknown groundwater pollution sources and estimation of aquifer parameters. Journal of Hydrology 376, 48–57 (2009)
7. Datta, B., Prakash, O., Campbell, S., Escalada, G.: Efficient Identification of Unknown Groundwater Pollution Sources using Linked Simulation-optimization incorporating Monitoring Location Impact Factor and Frequency Factor. Water Resources Management (2013), doi:10.1007/s11269-013-0451-8

8. Deb, K.: Multi-objective optimization using evolutionary algorithms, John Wiley and Sons, Ltd. (2001)
9. Dhar, A., Datta, B.: Saltwater Intrusion Management of Coastal Aquifers. I: Linked Simulation-Optimization. Journal of Hydrologic Engineering 14, 1263–1272 (2009)
10. Francone, F.D.: Discipulus^{TM} Software Owner's Manual, version 3.0 DRAFT. Machine Learning Technologies Inc., Littleton (1998)
11. Jha, M., Datta, B.: Three dimensional groundwater contamination source identification using adaptive simulated annealing. J. of Hydrologic Engineering 18(3), 307–313 (2012)
12. Koza, J.R.: Genetic programming as a means for programming computers by natural-selection. Statistics and Computing 4, 87–112 (1994)
13. Lin, J., Snodsmith, J.B., Zheng, C.M., Wu, J.F.: A modeling study of seawater intrusion in Alabama Gulf Coast, USA. Environmental Geology 57, 119–130 (2009)
14. Makkeasorn, A., Chang, N.B., Zhou, X.: Short-term streamflow forecasting with global climate change implications – A comparative study between genetic programming and neural network models. Journal of Hydrology 352, 336–354 (2008)
15. Prakash, O., Datta, B.: Optimal Monitoring Network Design for Efficient Identification of Unknown Groundwater Pollution Sources. Int. J. of GEOMATE 6(1) (sl. no. 11), 785–790 (2013, 2014)
16. Singh, R.M., Datta, B.: Identification of groundwater pollution sources using GA-based linked simulation optimization model. J. Hydrol. Eng. 11(2), 101–109 (2006)
17. 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)
18. Sreekanth, J., Datta, B.: Comparative Evaluation of Genetic Programming and Neural Network as Potential Surrogate Models for Coastal Aquifer Management. Water Resources Management 25, 3201–3218 (2011a)
19. Sreekanth, J., Datta, B.: Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate models and multiple-realization optimization. Water Resources Research 47 (2011b)
20. Sreekanth, J.: Integrated Multi-objective Management of Saltwater Intrusion in Coastal Aquifers Using Simulation-Optimization and Monitoring Feedback Information. Ph.D. Thesis, James Cook University (2012)

A Comparison of Fitness-Case Sampling Methods for Symbolic Regression with Genetic Programming

Yuliana Martínez[1], Leonardo Trujillo[1,⋆], Enrique Naredo[1], and Pierrick Legrand[2,3]

[1] TREE-LAB, 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
ysaraimr@gmail.com, leonardo.trujillo@tectijuana.edu.mx,
enriquenaredo@gmail.com
www.tree-lab.org

[2] Université Victor Segalen Bordeaux 2 and The Institut de Mathmatiques de Bordeaux, UMR CNRS 5251, France

[3] ALEA Team, INRIA Bordeaux Sud-Ouest, France
pierrick.legrand@u-bordeaux2.fr

Abstract. The canonical approach towards fitness evaluation in Genetic Programming (GP) is to use a static training set to determine fitness, based on a cost function averaged over all fitness-cases. However, motivated by different goals, researchers have recently proposed several techniques that focus selective pressure on a subset of fitness-cases at each generation. These approaches can be described as fitness-case sampling techniques, where the training set is sampled, in some way, to determine fitness. This paper shows a comprehensive evaluation of some of the most recent sampling methods, using benchmark and real-world problems for symbolic regression. The algorithms considered here are Interleaved Sampling, Random Interleaved Sampling, Lexicase Selection and a new sampling technique is proposed called Keep-Worst Interleaved Sampling (KW-IS). The algorithms are extensively evaluated based on test performance, overfitting and bloat. Results suggest that sampling techniques can improve performance compared with standard GP. While on synthetic benchmarks the difference is slight or none at all, on real-world problems the differences are substantial. Some of the best results were achieved by Lexicase Selection and Keep Worse-Interleaved Sampling. Results also show that on real-world problems overfitting correlates strongly with bloating. Furthermore, the sampling techniques provide efficiency, since they reduce the number of fitness-case evaluations required over an entire run.

Keywords: Fitness-Case Sampling, Symbolic Regression, Performance Evaluation.

1 Introduction

Genetic programming (GP) is a Machine Learning paradigm which, through an evolutionary process, tries to find the existence of implicit relationships within training data,

[⋆] Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_14, © Springer International Publishing Switzerland 2014

referred to as fitness-cases. Traditionally, GP algorithms use the entire training dataset to compute the fitness of each solution at every generation. However, recent works have reported improved performance when the entire training set is not used, and a different subset of fitness-cases are used at each generation instead, [1–3, 5, 8]. Several variants have been proposed, which will be discussed in the following sections, and successfully applied on various real-world problems. For instance, in [5] the authors use the entire set of fitness-cases in some generations, and use a randomly chosen subset in others. Another recent and related proposal is the Lexicase Selection algorithm [14], which is used as a means to evolve solutions to a specific conceptual class of problems called modal problems [15]. Modal problems are those for which solution programs must exhibit different modes of operations based on context. Lexicase selection is used to select parents based on different lexicographic orderings of fitness-cases, considering different orderings for each parent, and using the remaining fitness-cases to break ties. On the other hand, a behavior-based Novelty Search (NS) approach was used to solve symbolic regression problems in [10], where an explicit fitness function is not used to guide the search, instead a measure of uniqueness provides the selective pressure during evolution, biasing the search towards novel solutions. To this end, solutions are expressed by a descriptive vector called the $\epsilon - descriptor$, which describes the behavior of each individual based on the performance it exhibits on the entire training set. However, NS guides the search towards solutions that exhibit the best performance on the most difficult fitness-cases.

All of these approaches were motivated by different goals and assumptions, but basically produced quite similar algorithms, and share the same underlying strategy of focusing on a specific subset of fitness-cases to determine fitness. The present work is a comprehensive evaluation of the different fitness-case sampling techniques on benchmark and real-world problems, and presents a new fitness-case sampling technique based on Interleaved Sampling and the $\epsilon - descriptor$, called Keep-Worst Interleaved Sampling (KW-IS). Through experimental evaluation, insights are provided regarding the performance of these techniques on symbolic regression, based on test performance, overfitting and bloat.

This paper is organized as follows. Section 2 reviews previously proposed fitness-case sampling techniques. The KW-IS algorithm is described in Section 3. The experimental work and results are presented in Section 4. Finally, concluding remarks and future work are outlined in Section 5.

2 Fitness-Case Sampling Methods

Genetic Programming traditionally uses the entire set of fitness-cases to determine a program's fitness. However, some works have focused on evolving solutions using only a subset of fitness-cases, including the extreme configuration of only using a single one. The underlying goal of the proposed methods have been different, including reducing the amount of computation devoted to fitness [12], finding solutions that are not affected by bloat [6], avoiding overfitting [4, 5], analyzing problem modality [14], and finding novel solutions [7, 10]. Despite their diverse formulations, all of these methods share two basic similarities. First, all methods focus fitness assignment on a different subset of fitness-cases at different points during the search. Second, all methods should be able to steer the search towards the best possible performance for a given problem.

Therefore, the goal of this paper is to analyze and empirically compare these methods. However, we limit our study to those methods that enhance or modify the basic GP algorithm in very specific and bounded areas. For instance, [6, 12] promote fitness-case sampling within a coevolutionary framework, that makes it difficult to determine which aspect of the proposed algorithms are actually responsible for the differences in observed performance. Another example is the work in [7], where each fitness-case is weighted given its historical difficulty. However, in this case the approach has many possible variants, and was only tested with stack-based GP and boolean problems. Therefore, the following review focuses on methods with few degrees-of-freedom, that are implemented in a straightforward manner and easily combined with a basic tree-based GP with Koza-style search operators. In what follows, a review of each method is given, outlining their main features.

2.1 Interleaved Sampling and Related Methods

Interleaved Sampling (IS) [4] is a deterministic-based sampling method, which uses the entire training set to compute fitness in some generations and uses a single fitness-case in others. This approach was motivated by the idea of balancing learning and overfitting through the interleaving of fitness-cases, attempting to elude local optima. Determining in which generation to use a single fitness-case, and in which to use all of them, is an integral part of the algorithm. In [5], the authors present two variants that achieved the best results, IS and Random IS (RIS). The IS algorithm uses the entire training set to compute fitness every other generation, and uses a single randomly chosen fitness-case on the rest. RIS uses a probabilistic decision at the beginning of each generation, to determine if all of the fitness-cases are used or just a single random one. In [5], RIS exhibited the best performance with the probability of using a single fitness-case set to $p = 0.75$. These methods are related to the approach presented in [8], that proposes a new subset selection method that takes the problem structure into account, while also being problem independent.

2.2 Lexicase Selection

In [14] the authors propose the concept of modality to describe problems for which solutions must exhibit different modes of operations; i.e., solutions must exhibit distinct behaviors based on implicitly contextual information within the input data. Moreover, [14, 15] point out that, in general, GP systems are generally limited to problems for which solution programs can perform similar actions for all possible inputs, but real-world problems will normally require much more complex solutions. To solve such problems, [14] proposed the Lexicase Selection (LEX), a method for parent selection, that can allow each fitness-case to be the main source of selective pressure at any given parent selection event. During evolution, LEX selects parents by starting with a pool of candidates and filtering it on the basis of performance on single fitness-cases, considered one after another. In the basic implementation, the initial pool of candidates is the entire population and fitness-cases are ordered randomly. Moreover, only the best individuals are selected at each iteration, an elitist approach. LEX resembles a lexicographic character string ordering, where the first fitness-case has the largest effect on choosing the parent, then the next fitness-case acts as a tie-breaker, and so on. Initial experiments on a single symbolic regression problem for GP showed good results, but the method has not been tested on tree-based GP with standard benchmarks or real-world symbolic regression problems[1].

2.3 NS and the $\epsilon - descriptor$

In [10], the authors proposed a behavior-based strategy towards the search for solutions in symbolic regression. First, the authors proposed a method to describe how an individual solves a particular fitness-case, relative to the performance of the rest of the population, called the $\epsilon - descriptor$. In particular, to compute the $\epsilon - descriptor$, the proposed method determines which fitness-cases are the most difficult, based on the error of all the individuals within the population. Then, given N fitness-cases, the $\epsilon - descriptor$ for each individual is a binary vector of size N, where a value of 1 in the ith position means that the solution is in the top percentile of individuals within the population with respect to the ith fitness-case. The evolutionary process is then performed using the Novelty Search (NS) algorithm [9], which promotes solutions that are unique with respect to the rest of the population; in this case uniqueness is given by the corresponding $\epsilon - descriptor$ of each individual. Hence, at each generation, the search is biased towards solutions that exhibit the best performance on as many fitness-cases as possible, biasing the search towards difficult cases when the easiest fitness-cases are solved by most of the population. Moreover, recent work has suggested that a behavior-based NS algorithm is closely related to the concept of problem modality and the LEX method [15].

3 Keep-Worst Interleaved Sampling

Based on [10] and [5], this paper proposes a variant of both IS and the NS-based $\epsilon - descriptor$, which is here referred to as Keep-Worst Interleaved Sampling (KW-IS). KW-IS interleaves using the entire set of fitness-cases every other generation, just like IS would. However, in the remaining generations, fitness-cases are not chosen randomly. Instead, the goal is to bias selective pressure towards individuals that exhibit good performance on the most difficult fitness-cases. Therefore, the fitness-cases are ordered based on difficulty, given by the average error of the entire population in a given generation, and the $\rho\%$ most difficult fitness-cases are used to determine fitness in the next generation. In a series of informal evaluations, the best performance of this parameter was achieved with $\rho = 10\%$. As stated above, this proposal is based on the general methodology of the NS-based $\epsilon - descriptor$, but it is also common in other learning paradigms, such as AdaBoost for example, where solution design is adjusted based on

[1] Lexicase selection has mostly been applied using PushGP by its developers, with good results on a digital multiplier problem. Also, in a personal communication from the authors, it seems that it is also performing well on some specialized domains using tree-based GP, such as finding special terms in finite algebras.

the most difficult training data samples. Another example is [2], which describes some efforts made to reduce the number of such evaluations by selecting a small subset of difficult cases in the training data set on which to actually carry out the GP algorithm, however the authors employ a different heuristic sampling method to the one of KW-IS. Finally, KW-IS is also closely related to the historically assessed hardness approach of [7] developed for Push-GP. Indeed, KW-IS can be seen as a simplification or as a close variant of the algorithms described in [7]; however, in that work experimental tests were only performed on a single boolean problem.

4 Experiments

As stated before, a goal of this paper is to provide a comprehensive evaluation of the fitness-case sampling methods described above. In particular, we test IS, RIS, LEX and KW-IS on a set of five benchmark and three real-world symbolic regression problems. In the case of benchmarking, none of the algorithms have been extensively studied or compared. With regards to the real-world test problems, IS and RS have been previously evaluated in [5], but all other algorithms have not. Moreover, all algorithms are evaluated using a tree-based representation, with Koza-style crossover and mutation.

The algorithms are compared based on the following criteria. First, test set performance of the best solution found, a standard evaluation measure. Second is overfitting, which is here measured by the absolute difference between the training error of the best solutions and their respective test error. Finally, the algorithms are compared based on bloat, measured by the average size of the population in the final generation given by the number of nodes. All algorithms were implemented using the GPlab Matlab toolbox [13], and a standard GP (GP-STD) is included as a control algorithm for comparison. In all problem instances, a total of thirty independent runs were performed, with different training and testing sets. Results are presented as statistics over all runs, particularly the median, mean and standard error. For each problem, box plots are used to show the variance of the runs and median performance, while statistical measures are computed excluding outliers.

4.1 Benchmark Problems

Five benchmark problems are used, originally published in [16], and suggested in [11] and [10]; these problems are summarized in Table 1. The experimental parameters used with these problems are given in Table 2, and fitness is calculated as the root mean square error between predicted and expected outputs.

The five box plots in Figure 1 show the performance of the best individual from each run evaluated with the test set. Figure 2, on the other hand, shows the overfitting results for each method. Figure 3 summarizes the bloat results in the final generation. A numerical comparison of the methods is provided in Table 3, showing in bold the best result based on each measure.

On these benchmark problems the following trends are apparent. It seems that test performance is mostly equivalent across all methods, with only slight advantages in some problems for some methods. Probably the most interesting result is that GP-STD is actually quite competitive, achieving similar results to the sampling techniques.

Table 1. Benchmark symbolic regression problems

Problem	Function	Fitness/Test cases
f_1 Benchmark	$x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
f_2 Benchmark	$x^5 + x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
f_3 Benchmark	$sin(x^2) * cos(x) - 1$	20 random points $\subseteq [-1, 1]$
f_4 Benchmark	$log(x + 1) + log(x^2 + 1)$	20 random points $\subseteq [0, 2]$
f_5 Benchmark	$2sin(x) * cos(y)$	100 random points $\subseteq [-1, 1]x[-1, 1]$

Table 2. GP Parameters used with the benchmark problems

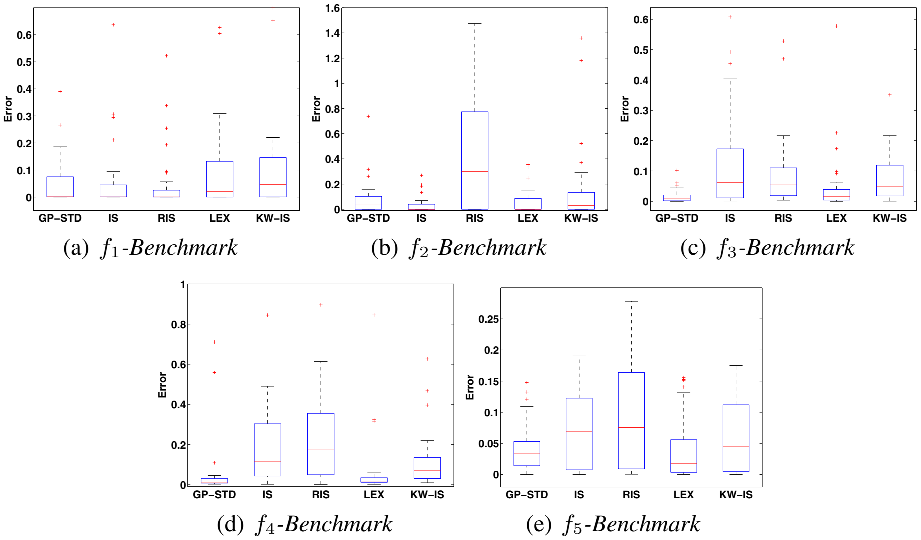
Parameter	Description
Population size	200 individuals
Generations	100 generations
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth
Operator probabilities	Crossover $p_c = 0.9$, Mutation $p_\mu = 0.05$
Function set	$(+ , - , \times , \div , sin , cos , exp , log)$.
Terminal set	$x, 1$ for single variable problems and x, y for bivariable problem.
Maximum tree depth	20 levels
Selection	Tournament selection of size 3
Elitism	Best individual always survives

Moreover, based on overfitting, once again GP-STD is quite good on these problems, which was not expected based on previously published works. In fact, the only method that shows overfitting on these problems is RIS. It is also noteworthy that LEX also exhibits competitive performance, even though its original design was not intended for these type of problems. Regarding the phenomenon of bloat all techniques show a similar trend to evolve populations of similar average size.

4.2 Real-World Problems

To get a better assessment of each method in more difficult scenarios, the three problems described in [5] are used to evaluate the algorithms; these are: Toxicity, Plasma Protein Binding and Bioavailability, which are characterized by a high-dimensionality and an extremely difficult to model behavior. The experimental parameters used are the same as those in [5], summarized in Table 4. Fitness is calculated as the root mean square error between predicted and expected outputs, and the data set is randomly divided before each run, using 50% for training and 50% for testing. Figure 4 presents box plots of performance on the test data and Figure 5 summarizes the overfitting results. Bloating of each method is shown in Figure 6, and a numerical comparison of the methods is provided in Table 5.

Given the uniqueness of each problem, we evaluate performance on each separately. First, for the Toxicity problem, based on median and mean test error LEX achieves



(a) f_1-*Benchmark* (b) f_2-*Benchmark* (c) f_3-*Benchmark*

(d) f_4-*Benchmark* (e) f_5-*Benchmark*

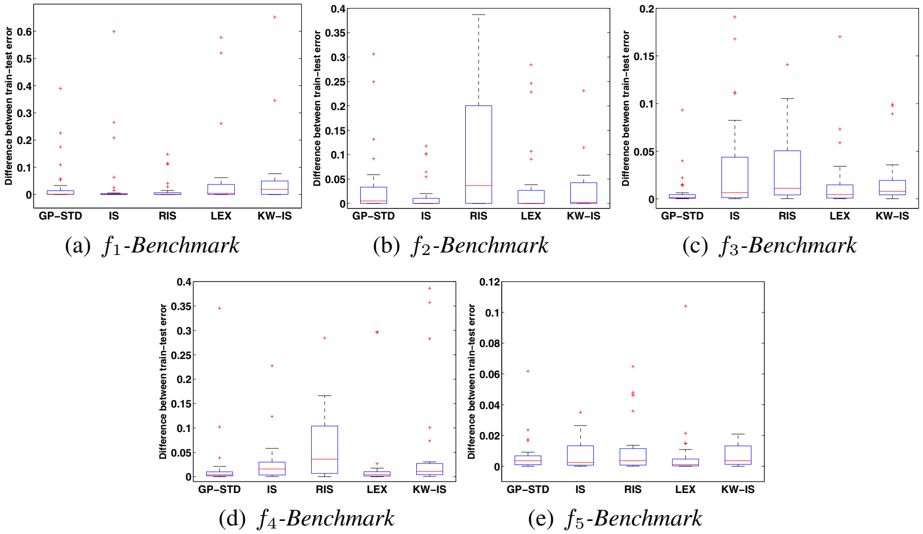
Fig. 1. The box plots show test error of the best individual over all thirty runs, on each problem

slightly better results, but all methods are competitive. However, GP-STD exhibited the worst performance on this problem in our runs, while IS, RIS and KW-IS are very similar. Based on overfitting, again GP-STD exhibited the worst performance on this problem, and IS, RIS, LEX and KW-IS show almost no overfitting. Finally, based on bloat, GP-STD evolves large trees during the evolutionary process, on the other hand IS, RIS, LEX and KW-IS control bloating more effectively.

Second, considering the Plasma Protein Binding problem. Based on test error, KW-IS achieved the best results, followed by IS and LEX, and the worst results are obtained by GP-STD and RIS. Overfitting on this problem was only exhibited by GP-STD and all sampling methods achieved strong results. Finally, GP-STD exhibited a high degree of bloating, while IS, RIS LEX and KW-IS controlled bloat effectively.

The final problem is Bioavailability, and the results suggest the following. Testing error shows that LEX gives the best results, with KW-IS close behind, while GP-STD, IS and RIS achieve the worst results. On the other hand, GP-STD again showed the most overfitting, all other sampling techniques do not show any substantial overfitting. Regarding bloat, GP-STD once again bloats, while all other methods do not.

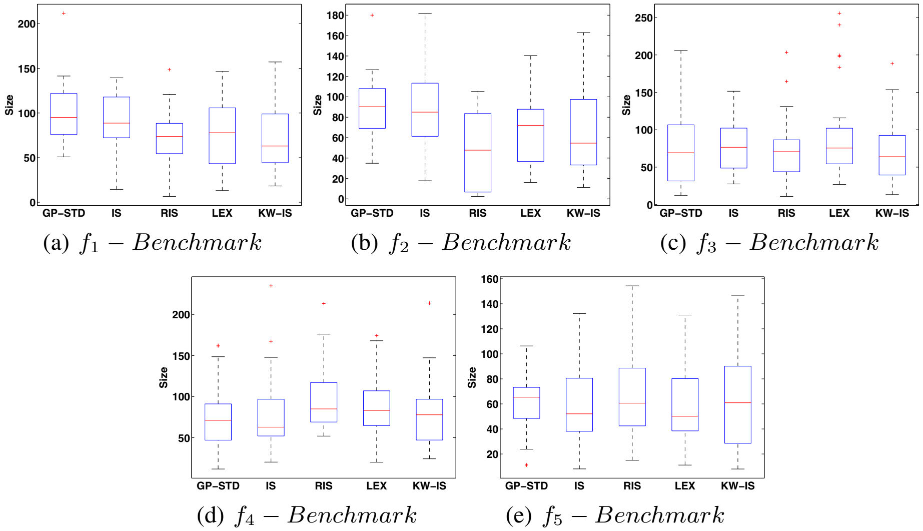
In summary, it appears that based on performance, IS, LEX and KW-IS achieve very good results, and GP-STD is actually also quite good, which was unexpected. However, IS, RIS, LEX and KW-IS do not overfitt and GP-STD exhibit substantial amounts of overfitting. IS, RIS, LEX and KW-IS seems to achieve a good compromise between performance and overfitting, without incurring bloat.



(a) f_1-*Benchmark* (b) f_2-*Benchmark* (c) f_3-*Benchmark*

(d) f_4-*Benchmark* (e) f_5-*Benchmark*

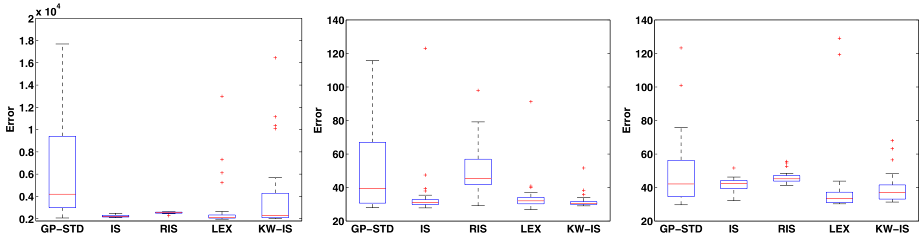
Fig. 2. The box plots shows the difference between the error for training data and error for test data, we see whether or not we are overfitting



(a) $f_1 - Benchmark$ (b) $f_2 - Benchmark$ (c) $f_3 - Benchmark$

(d) $f_4 - Benchmark$ (e) $f_5 - Benchmark$

Fig. 3. Box plots of average program size in the final generation as a measure of bloat



(a) *Toxicity* (b) *Plasma Protein Binding* (c) *Bioavailability*

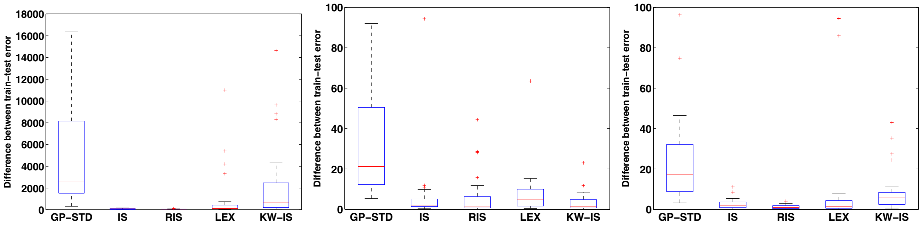
Fig. 4. The box plots show test error of the best individual from each run

Table 3. The median, mean, and standard error achieved by each method on the five benchmark problems; bold indicates best

Problem	GP-STD	IS	RIS	LEX	KW-IS
			Median		
f1-Benchmark	0.0031	**0.0000**	**0.0000**	0.0207	0.0468
f2-Benchmark	0.4810	**0.0000**	0.2990	**0.0000**	0.0286
f3-Benchmark	**0.0079**	0.0616	0.0569	0.0166	0.0496
f4-Benchmark	**0.0126**	0.1172	0.1733	0.0180	0.0691
f5-Benchmark	0.0342	0.0695	0.0756	**0.0180**	0.0455
			Mean		
f1-Benchmark	0.0574	0.0599	**0.0526**	0.2341	0.1972
f2-Benchmark	0.0841	**0.0352**	3.7409	0.0572	0.1617
f3-Benchmark	**0.0163**	0.1248	0.0919	0.0522	0.0799
f4-Benchmark	0.7990	0.1806	0.2261	**0.0681**	0.1469
f5-Benchmark	**0.0400**	0.0693	0.1060	0.0457	0.0781
			Standard Deviation		
f1-Benchmark	**0.0935**	0.1372	0.1213	0.6598	0.4088
f2-Benchmark	0.1450	**0.0689**	18.408	0.0983	0.3284
f3-Benchmark	**0.0226**	0.1630	0.1252	0.1121	0.0853
f4-Benchmark	4.0420	0.1831	0.2088	**0.1659**	0.2196
f5-Benchmark	**0.0410**	0.0615	0.1372	0.0556	0.1123

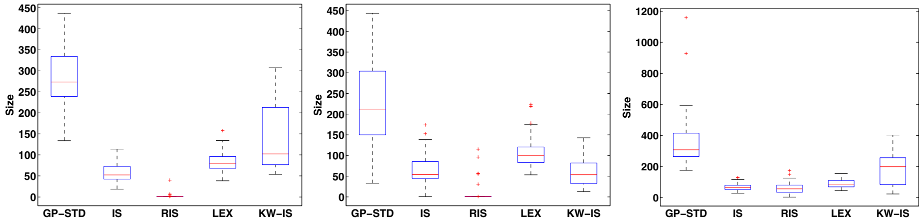
Table 4. Parameters for the symbolic regression real-life problems in GP

Parameter	Description
Population size	500 individuals
Generations	200 generations
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth
Operator probabilities	Crossover $p_c = 0.9$, Mutation $p_\mu = 0.05$
Function set	$\{+, -, \times, \div\}$
Terminal set	Input variables, constants $-1.0, -0.5, 0, 0.5$ and 1.0
Maximum tree depth	17 levels
Selection	Tournament selection of size 10
Elitism	Best individual always survives



(a) *Toxicity* (b) *Plasma Protein Binding* (c) *Bioavailability*

Fig. 5. The box plots shows the difference between training and testing error as an estimate of overfitting



(a) *Toxicity* (b) *Plasma Protein Binding* (c) *Bioavailability*

Fig. 6. Box plots of average program size in the final generation as a measure of bloat

Table 5. The median, mean, and standard error achieved by each method on the three real-world problems; bold indicates best

Median

Problem	**GP-STD**	**IS**	**RIS**	**LEX**	**KW-IS**
Toxicity	3639.20	2217.20	2555.50	**2085.30**	2210.50
Plasma Protein Binding	37.90	30.70	45.03	31.43	**30.20**
Bioavailability	39.02	42.22	44.65	**33.17**	35.42

Mean

Toxicity	4618.30	2232.80	2537.60	**2134.60**	2613.60
Plasma Protein Binding	47.10	**31.01**	50.38	31.52	30.61
Bioavailability	41.53	41.64	44.97	**34.07**	36.86

Standard Deviation

Toxicity	3316.90	99.89	**73.73**	155.80	946.04
Plasma Protein Binding	22.91	1.94	13.08	2.42	**1.11**
Bioavailability	10.54	3.23	**1.89**	3.47	4.79

5 Conclusions

This work reviews GP methods that were designed to focus the search on different fitness-cases at each generation, by sampling the set of fitness-cases in different ways. In particular, the methods called Interleaved Sampling, Random Interleaved Sampling and Lexicase Selection were discussed, as well as Novelty Search based $\epsilon - descriptor$ method for symbolic regression. After reviewing these methods, a new sampling algorithm is proposed, which evaluates individuals using the most difficult set of fitness-cases every other generation, as determined by the population from the previous generation.

These algorithms were evaluated using benchmark problems suggested in [11] and real-world problems studied recently in [5]. Results on benchmark problems show a similar behavior in terms of performance, overfitting and bloat. However, in real-world problems, it is observed that GP-STD achieves the worst performance of all methods. On these problems, it seems that varying the training set, in some way, helps produce better solutions. Moreover, among all sampling methods tested here, Lexicase Selection produced the best results, with KW-IS close behind.

A noteworthy insight is the fact that on the more difficult real-world problems, overfitting seems to be correlated with bloat, since standard GP exhibits both, while the sampling techniques are able to eliminate both. These results are interesting, based on previous works that suggest that overfitting and bloat should be considered as separate phenomena [17]. Finally, future work will focus on evaluating these methods on a wider range of problem instances and domains, and on developing new techniques that are able to improve upon the results presented here.

Acknowledgments. Funding provided by CONACYT (Mexico) Basic Science Research Project No. 178323, DGEST (Mexico) Research Projects No.5149.13-P and TIJ-ING-2012-110, and IRSES project ACoBSEC from the European Commission. First and third authors supported by CONACYT (Mexico) scholarships, respectively No. 226981 and No. 232288.

References

[1] Doucette, J., Heywood, M.I.: Gp classification under imbalanced data sets: Active subsampling and auc approximation. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 266–277. Springer, Heidelberg (2008)

[2] Gathercole, C., Ross, P.: Dynamic training subset selection for supervised learning in genetic programming. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 312–321. Springer, Heidelberg (1994)

[3] Giacobini, M., Tomassini, M., Vanneschi, L.: Limiting the number of fitness cases in genetic programming using statistics. 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. 371–380. Springer, Heidelberg (2002)

[4] Gonçalves, I., Silva, S.: Experiments on controlling overfitting in genetic programming. In: 15th Portuguese Conference on Artificial Intelligence (EPIA 2011) (October 2011)

[5] Gonçalves, I., Silva, S.: Balancing learning and overfitting in genetic programming with interleaved sampling of training data. In: Krawiec, K., Moraglio, A., Hu, T., Etaner-Uyar, A.Ş., Hu, B. (eds.) EuroGP 2013. LNCS, vol. 7831, pp. 73–84. Springer, Heidelberg (2013)

[6] Harper, R.: Spatial co-evolution: Quicker, fitter and less bloated. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, GECCO 2012, pp. 759–766. ACM, New York (2012)

[7] Klein, J., Spector, L.: Genetic programming with historically assessed hardness. In: Genetic Programming Theory and Practice VI. Genetic and Evolutionary Computation, pp. 1–14. Springer US (2009)

[8] Lasarczyk, C.W.G., Dittrich, P.W.G., Banzhaf, W.W.G.: Dynamic subset selection based on a fitness case topology. Evol. Comput. 12(2), 223–242 (2004)

[9] Lehman, J., Stanley, K.O.: Exploiting open-endedness to solve problems through the search for novelty. In: Proceedings of the Eleventh International Conference on Artificial Life, Cambridge, MA, ALIFE XI. MIT Press (2008)

[10] Martínez, Y., Naredo, E., Trujillo, L., López, E.G.: Searching for novel regression functions. In: IEEE Congress on Evolutionary Computation, pp. 16–23 (2013)

[11] McDermott, J., White, D.R., Luke, S., Manzoni, L., Castelli, M., Vanneschi, L., Jaskowski, W., Krawiec, K., Harper, R., De Jong, K., O'Reilly, U.-M.: Genetic programming needs better benchmarks. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, GECCO 2012, pp. 791–798. ACM, New York (2012)

[12] Schmidt, M., Lipson, H.: Coevolving fitness models for accelerating evolution and reducing evaluations. In: Riolo, R., Soule, T., Worzel, B. (eds.) Genetic Programming Theory and Practice IV. Genetic and Evolutionary Computation, pp. 113–130. Springer US (2007)

[13] Silva, S., Almeida, J.: Gplab–a genetic programming toolbox for matlab. In: Gregersen, L. (ed.) Proceedings of the Nordic MATLAB Conference, pp. 273–278 (2003)

[14] Spector, L.: Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion, GECCO Companion 2012, pp. 401–408. ACM (2012)

[15] Trujillo, L., Spector, L., Naredo, E., Martínez, Y.: A behavior-based analysis of modal problems. In: GECCO (Companion), pp. 1047–1054 (2013)

[16] Uy, N.Q., Hoai, N.X., O'Neill, M., Mckay, R.I., Galván-López, E.: Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Genetic Programming and Evolvable Machines 12(2), 91–119 (2011)

[17] Vanneschi, L., Castelli, M., Silva, S.: Measuring bloat, overfitting and functional complexity in genetic programming. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010, pp. 877–884. ACM, New York (2010)

Evaluating the Effects of Local Search in Genetic Programming

Emigdio Z-Flores[1], Leonardo Trujillo[1,⋆], Oliver Schütze[2], and Pierrick Legrand[3,4]

[1] TREE-LAB, 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
emigdio.zflores@tree-lab.org,
leonardo.trujillo@tectijuana.edu.mx
www.tree-lab.org

[2] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, Col. San Pedro Zacatenco, 07360, Mexico City, México,
schuetze@cs.cinvestav.mx

[3] Université Victor Segalen Bordeaux 2 and The Institut de Mathématiques de Bordeaux, UMR CNRS 5251, France

[4] ALEA Team, INRIA Bordeaux Sud-Ouest, France
pierrick.legrand@u-bordeaux2.fr

Abstract. Genetic programming (GP) is an evolutionary computation paradigm for the automatic induction of syntactic expressions. In general, GP performs an evolutionary search within the space of possible program syntaxes, for the expression that best solves a given problem. The most common application domain for GP is symbolic regression, where the goal is to find the syntactic expression that best fits a given set of training data. However, canonical GP only employs a syntactic search, thus it is intrinsically unable to efficiently adjust the (implicit) parameters of a particular expression. This work studies a Lamarckian memetic GP, that incorporates a local search (LS) strategy to refine GP individuals expressed as syntax trees. In particular, a simple parametrization for GP trees is proposed, and different heuristic methods are tested to determine which individuals should be subject to a LS, tested over several benchmark and real-world problems. The experimental results provide necessary insights in this insufficiently studied aspect of GP, suggesting promising directions for future work aimed at developing new memetic GP systems.

Keywords: Genetic Programming, Local Search, Memetic Algorithms.

1 Introduction

It is widely understood that effectively balancing exploration and exploitation is one of the key issues underlying any successful search process. Indeed, a useful taxonomy of search algorithms is based on the idea of differentiating between search methods that rely on exploratory search operators, and search methods that can guarantee (local)

[⋆] Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_15, © Springer International Publishing Switzerland 2014

optimality by exploiting the local structure of a fitness landscape. Local search (LS) algorithms can guarantee convergence to local optima if their underlying assumptions are satisfied and are computationally efficient. However, the success of a LS strongly depends on the initial point, particularly in highly irregular, multimodal or discontinuous fitness landscapes [10]. On the other hand, global search algorithms include a variety of deterministic and stochastic strategies. Here, we focus on evolutionary algorithms (EAs), metaheuristic strategies based on an abstract model of Neo-Darwinian evolution [7, 19]. EAs are particularly useful when a good initial solution is difficult to propose, or when LS strategies tend to converge on undesirable local optima. Moreover, EAs are particularly well suited when a single monolithic solution is not sufficient, and instead a set of different solutions is required [2, 6]. In general, EAs cannot guarantee convergence towards local optima in most realistic scenarios. Moreover, they tend to be computationally costly, and in many cases can be highly inefficient. Nevertheless, EAs have produced extremely competitive solutions in difficult domains and problem instances [15].

Considering the strengths and weaknesses of both global and local methods, it is intuitive to conclude that the best strategy should be a hybrid approach, commonly referred to as memetic search. These methods have been extensively studied over recent years, combining EAs with a variety of local searchers [1].

While all EAs are based on the same general principles [5], there is a large variety among current methods, each with their respective strengths and weaknesses. This paper focuses on an EA paradigm that presents unique properties among other EAs, genetic programming (GP) [16, 25]. The goal of GP is to automatically evolve specialized syntactic expressions that best solve a given tasks, which can be interpreted as mathematical functions or computer programs. In particular, this paper studies how the standard syntactic search can be improved by including a LS method. While this paper is not the first to develop a memetic GP system, it does present the first systematic evaluation of what could be the best strategies to accomplish this, by considering several variants and evaluating them on current benchmark problems for symbolic regression.

The remainder of this paper proceeds as follows. First, an overview of GP is provided in Section 2. Then, Section 3 reviews previous work on memetic optimization with EAs. Afterwards, Section 4 describes our proposed memetic algorithms to perform real-valued parameter optimization of evolved syntactic expressions, considering several basic variants. The experimental setup is presented in Section 5 and results are discussed in Section 6. Finally, a summary of the paper and concluding remarks are outlined in Section 7.

2 Genetic Programming

GP can be understood as a generalization of the basic genetic algorithm, its main features can be summarized as follows [25]. First, GP was originally proposed as an EA that evolves simple programs, functions, operators, or in general symbolic expressions that perform some form of computation. GP is basically used to evolve solutions to different types of design problems, with examples as varied as quantum algorithms [31], computer vision operators [23] and satellite antennas [12]. Second, solutions are expressed as variable length structures, such as linked lists, parse trees or graphs [25].

These structures encode the syntax of each individual program. Therefore, in a canonical GP algorithm, search operators, such as crossover and mutation, perform syntactic variations on the evolving population. Third, by considering each individual in a GP run as a program, the evolutionary process is basically attempting to write the best program syntax that solves a given problem. Therefore, a finite set of basic symbols needs to be defined, which is called the primitive set $\mathbb{P}$. Within the primitive set there is a subset of basic operations or functions of different arity, called the function set F, and a subset of input variables, constants or zero arity functions called the terminal set T, such that $\mathbb{P} = F \cup T$.

Given the variety of possible GP configurations and applications, this work focuses on the problem of symbolic regression using a tree representation. In symbolic regression, the goal is to search for the symbolic expression $K^O : \mathbb{R}^p \rightarrow \mathbb{R}$ that best fits a particular training set $\mathbb{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of n input/output pairs with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. In such problems, for instance, the function set can be defined as $F = \{+, -, /, *\}$ and the terminal set can be composed by each of the features in the input data, such that $T = \{x_i\}$ with $i = 1, \ldots, p$, but other terminal elements can be included such as integer or real-valued constants.

The general symbolic regression problem can then be defined as

$$(K^O, \boldsymbol{\theta}^{\boldsymbol{O}}) \leftarrow \underset{K \in \mathbb{G}; \boldsymbol{\theta} \in \mathbb{R}^m}{arg\,min} \; f(K(\mathbf{x}_i, \boldsymbol{\theta}), y_i) \; with \; i = 1, \ldots, p \,, \tag{1}$$

where $\mathbb{G}$ is the solution or syntactic space defined by $\mathbb{P}$, f is the fitness function based on the difference between a program's output $K(\mathbf{x}_i, \boldsymbol{\theta})$ and the expected output y_i, such as the mean square error, and $\boldsymbol{\theta}$ is a particular parametrization of the symbolic expression K, assuming m real-valued parameters. This dual problem, of simultaneously optimizing syntax (structure) as well as its parametrization is also discussed in [8, 18]. The authors differentiate between two possible approaches towards solving such a task. The first group is *hierarchical structure evolution* (HSE), when $\boldsymbol{\theta}$ has a strong influence on fitness, and thus a local search is required at each generation of the global (syntactic) search, configured as a nested process. The second group is called *simultaneous structure evolution* (SSE), when $\boldsymbol{\theta}$ has a marginal effect on fitness, in such cases a single evolutionary loop could optimize both syntax and parameters simultaneously. However, such categorizations are highly abstract, and a particular implementation could easily be classified into both groups. Nevertheless, it is reasonable to state that the standard GP approach falls in the SSE group.

In standard GP, however, parameter optimization is usually not performed explicitly, since GP search operators only affect syntax. Therefore, the parameters are only implicitly considered. For instance, a GP individual might have the following syntax $K(x) = x + sin(x)$; in this case, we might propose the following parametrization: $\boldsymbol{\theta} = (\alpha_1, \alpha_2, \alpha_3)$, with $K(x) = \alpha_1 x + \alpha_2 sin(\alpha_3 x)$. In a traditional GP search, these parameters are all set to 1, which does not necessarily lead to the best possible performance for this particular syntax. Indeed, if the optimal solution is

$K^O(x) = 3.3x + 1.003sin(0.0001x)$, then individual K might be easily lost during the selection or survival processes[1].

This seems to be a glaring weakness in the standard GP approach. While the syntactic-based search in GP has solved a variety of difficult problems, it is nonetheless very inefficient, leading to important practical limitations, such as search stagnation, bloat and uninterpretable solutions [28]. In other words, GP performs a highly explorative search, since the search operators can produce large fitness changes with only modest syntactic modifications or vice-versa. Therefore, the inclusion of a LS procedure could enhance the performance of the search. In this paper, the goal is to study what are the effects of including a local optimization strategy within a GP run, configured as an HSE system.

3 Previous Work

As stated above, many works have studied how to combine an EA with a local optimizer (also referred to as a refinement process). In general, such approaches are considered to be a simple type of memetic search [1]. The basic idea is straightforward: include within the optimization process an additional search operator that takes an individual (or several) as an initial point and searches for the local optima around it. Such a strategy can help guarantee that the local region around each individual is fully exploited. However, there can be some negative consequences to such an approach. The most evident is computational overhead, while the cost of a single LS might be negligible, performing it on every individual might become inefficient. Second, LS can produce overfitted solutions, stagnating the search on local optima. These issues aside, hybrid techniques have produced impressive results in a variety of scenarios [1].

A useful taxonomy of this type of memetic algorithms can be derived based on how inheritance is carried out during the evolutionary process [1]. Suppose that $\mathbf{h}, \mathbf{h}^o \in \mathbb{G}$, where $\mathbf{h}$ is an individual solution and $\mathbf{h}^o$ is the solution generated after a LS is applied on $\mathbf{h}$. Obviously, for a minimization problem, $f(\mathbf{h}^o) \leq f(\mathbf{h})$, with f the objective function. Thus, $\mathbf{h} \neq \mathbf{h}^o$, unless $\mathbf{h}$ was in fact a local optima. Then, a memetic algorithm could proceed in two distinct ways with respect to inheritance. In a *Lamarckian* algorithm, the traits acquired during the local search, captured in $\mathbf{g}^o$, replace those of the original individual $\mathbf{h}$; i.e., the inheritance of acquired characteristics $(\mathbf{h}, f(\mathbf{h})) \rightarrow (\mathbf{h}^o, f(\mathbf{h}^o))$. On the other hand, in a *Baldwinian* algorithm, the local optimization process only modifies the fitness of an individual; $(\mathbf{h}, f(\mathbf{h})) \rightarrow (\mathbf{h}, f(\mathbf{h}^o))$; i,e., ontogenic evolution.

An extensive survey of these methods is presented in [1] by Chen et al.. A noteworthy aspect of this survey is an almost complete lack of papers that deal with GP. Of the more than two hundred papers covered in [1], only a couple deal with memetic GP. This illustrates how the GP community has not addressed the topic adequately. Nonetheless, some works have been developed over recent years. For instance, [38] presents

[1] Some GP systems do include numerical terminals or random ephemeral constants, while these constants can act as coefficients in an evolved expressions, they are not subject to optimization after they are introduced into the syntax of a particular individual.

a memetic GP approach to evolve decision trees. The authors report good results using domain-specific LS heuristics. In [9] the authors present a memetic crossover for GP, instead of a local search strategy, crossover is based on a more general notion of memetic search.

Indeed, the complete optimization problem defined in Section 2 has not received much attention. In [33], gradient descent is used to optimize numerical constants within a GP tree, achieving good results on five symbolic regression problems. However, the work only optimizes the value of numerical terminal elements (leaves), it does not consider parameters within internal nodes. Additionally, the paper presents results from a small sample size of runs, and only considers training fitness, a highly deceptive measure of learning. Similarly, in [42] and [11] a LS algorithm is used to optimize the value of constant terminal elements. In [42] gradient descent is used and tested on classification problems, while [11] uses RPROP and evaluates the proposal on a real-world problem, in both cases leading towards improved results. While these works show promise, they include several design choices that are not analyzed or justified in detail. For instance, [42] applies gradient descent on every individual of the evolving population, an obvious computational bottleneck, while [11] only applies RPROP on the best individual from each generation. A similar strategy is included in the HeuristicLab optimization environment, where local optimization can be performed on the final solution found by GP [37]. In these cases, it is not evident which proposed strategy can offer the best results in new scenarios. Moreover, [11, 42] only evaluate their approaches on specific problem instances.

Closer to the problem discussed here, [29] includes weight parameters for each function node, what the authors call inclusion factors; these weights modulate the importance that each node has within the tree. Indeed, the authors identify what we are here referring to as implicit program parameters, and optimize these values by applying gradient descent on all trees. The authors also propose a series of new search operators that explicitly contemplate the parametrization of each GP tree. However, only a limited experimental validation is performed on specialized classification problems, with mixed results. The performance of the memetic GP is indeed better, but not substantially in some tests. Additionally, the improvement in performance is misleading, since the GP systems are executed a fixed number of generations, but the added computational search performed by gradient descent leads to an unfair comparison based on total search effort. Moreover, it seems that the proposed parameter aware search operators have a negative effect on the search. The GP system is evaluated using an uncommonly small function set (only 2 functions), an unrealistic configuration.

Finally, recent works have decided to completely change the basic GP framework in order to account for the lack of an explicit parametrization of syntactic expressions. The fast function extraction (FFX) algorithm [20], for instance, poses the symbolic regression problem as that of finding a linear combination of a subset of candidate basis functions. Thus, FFX builds linear in parameter models, and optimizes using a modified version of the elastic net regression technique, eliminating the evolutionary process altogether. Another recent example is the prioritized grammar enumeration (PGE) technique [40], that employs dynamic programming and eliminates the basic search operators of traditional GP. Parameter values of the symbolic expressions produced by PGE are optimized using non-linear optimization with the Levenberg-Marquardt algorithm.

4 Integrating Local Optimization Strategies within GP

Intuitively, through tree parametrization and local optimization, a GP search should converge faster towards high quality solutions. As stated before, the proposal is to develop an HSE-GP, by parameterizing GP trees and including a Lamarckian memetic strategy. Therefore, the parametrization of GP trees must be defined; then, a particular LS method must be chosen. Finally, a decision strategy must be suggested to determine on which individuals should a LS be applied. The proposals for each of these issues are presented below.

4.1 Parametrization of GP Trees

For this study, we propose a simple and naive approach to add parameters within GP trees. For each function in the function set $g_k \in F$, we add a unique weight coefficient $\theta_k \in \mathbb{R}$, such that each function is now defined by

$$g'_k = \theta_k g_k \tag{2}$$

where g'_k is the new parameterized function, $k \in \{1, \ldots, r\}$ and $r = |F|$. Note that each θ_k is linked to a single g_k, such that the parameter vector of tree K_i is given by $\boldsymbol{\theta} \in \mathbb{R}^m$ with m the number of different functions included in K_i such that $m \leq r$. Notice that if a tree contains several instances of a particular function, all instances of this function share the same coefficient. Indeed, this severely constraints the optimization process, particularly for large trees that can include many instances of the same function. To be clear, it is not argued that such a parametrization should be considered as the best possible alternative. Nonetheless, it does have one important consequence, it bounds the size of the search space for each LS process, something that could become overwhelming if the GP trees grew to large, which tends to happen frequently due to bloat [28].

In all trees, every θ_k is initialized to unity, which would be their implicit value in a standard GP. Since the proposed algorithm is a Lamarckian memetic search, the standard GP search operators (subtree mutation and subtree crossover) still only operate at the syntax level, exchanging g'_k nodes without affecting their respective θ_k.

4.2 Local Search Mechanism

Potentially, any tree can be of linear or non-linear form; however, for convenience we treat every tree as non-linear expression. This is a multidimensional non-linear optimization problem, which can be solved using least squares.

The above problem, is formally expressed by the following cost function

$$\underset{\boldsymbol{\theta}}{min} \|K(\boldsymbol{\theta}, \mathbf{x}) - \mathbf{y}\|_2^2 = \underset{\boldsymbol{\theta}}{min} \sum_i (K(\boldsymbol{\theta}, \mathbf{x_i}) - \mathbf{y_i})^2 \ , \tag{3}$$

where $\mathbf{x}$ are the input data points, $\mathbf{y}$ are the output data points, i is the index for the regression instances, and K is the non-linear function. The problem posed in (3) can be solved using different methods [41] [17]. In this case, we chose to use a well known technique with good convergence properties and good scalability in complexity for high dimensional problems, called trust region optimization [30].

The trust region optimization method tries to minimize a smooth non-linear function subject to bounds on the variables, given by

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^m}{min} K(\boldsymbol{\theta}),\ l_i \leq \theta_i \leq u_i\ \forall i \in \{1..m\}\ , \tag{4}$$

where $l_i \in \{\mathbb{R} \cup \{-\infty\}\}$, $u_i \in \{\mathbb{R} \cup \{\infty\}\}$, and $K : \mathbb{R}^m \to \mathbb{R}$. Conceptually, a trust region approach replaces an m-dimensional unconstrained optimization problem by an m-dimensional constrained problem. This results in an approximate solution, since it does not needed to be solved with high accuracy. One of the appealing points of these methods is their strong convergence properties [3]. The idea behind a trust region method for $min_{\theta\in\mathbb{R}^m} K(\theta)$ is simple. The increment $s_k = x_{k+1} - x_k$ is an approximate solution to a quadratic subproblem with a bound on the step size

$$\underset{s\in\mathbb{R}^m}{min} \left\{ \psi_k(s) \overset{def}{=} g_k^T s + \frac{1}{2} s^T B_k s : \|\overline{D}_k\| \leq \mathit{\Delta}_k \right\}\ , \tag{5}$$

where $g_k \overset{def}{=} \nabla f(x_k)$, B_k is a symmetric aproximation to the Hessian matrix $\nabla^2 f(x_k)$, $\overline{D}_k$ is a scaling matrix, and $\mathit{\Delta}_k$ is a positive scalar representing the trusted region size. Solving (5) efficiently is not a trivial task, see [22,26,30,32]. Here, the method proposed in [4] is used, which does not require the solution of a general quadratic programming subproblem at each iteration.

4.3 Integrating LS into GP

The final issue that must be considered is to determine on which individuals is the LS applied, and at what moment during the evolutionary process. For the latter point, LS is applied after a complete evolutionary loop is completed; i.e., LS is applied after the survival criterion is applied and before the following generation begins. This means that the local optimization of individuals from generation t impacts the search at generation $t + 1$. For the former point, several different approaches are evaluated.

First, three naive approaches are considered, two of which have been used in previous memetic GP systems. Probably the simplest is to apply a LS on all of the individuals, this method is referred to as LSALL-GP. This approach will inherently introduce a large computational overhead. Another approach is to be more selective, and only apply a LS on the best individual from each generation, this method is referred to as LSBEG-GP. Conversely, LS can also be applied on the worst individual from each generation, this variant is called LSWEG-GP. This variant might be useful for extreme cases of individuals that present the correct structure, but a highly suboptimal parametrization.

The fourth variant is referred to LS Random Population or LSRP-GP, where every individual of the population is a viable candidate for a LS optimization with a probability p_{LS}, basically implemented as a mutation operator. Here, it is assumed that a

Table 1. GP Parameters for the different methods

Parameter	LSBEG-GP	LSWEG-GP	LSRP-GP	LSBS-GP	LSWS-GP	LSALL-GP
θ	vector of ones					
F_{BEST}	yes	no	no	no	no	no
F_{WORST}	no	yes	no	no	no	no
p_{LS}	1	1	0.1	0.5	0.5	1
PER_{LS}	0	0	100%	10%	10%	100%

low probability is desirable, to minimize the added computational cost. The fifth variant is called LS Best Subset or LSBS-GP, which is the same as LSRP-GP, except that the individuals that are valid candidates for LS are those in the best PER_{LS} percentile of the population, a second algorithm parameter. Finally, the sixth variant is called LS Worst Subset or LSWS-GP, takes the same approach as LSRP-GP, but candidate individuals for a LS are those in the worst PER_{LS} percentile. All of the methods and their parameters are summarized in Table 1.

5 Experimental Setup

The algorithms that are evaluated are LSBEG-GP, LSWEG-GP, LSRP-GP, LSBS-GP, LSWS-GP and LSALL-GP; also, a standard GP is included as a control method. All the algorithms were implemented in Matlab using the GPLAB[2] Toolbox [27] modifying its core functionality to integrate the LS procedure. The set of experiments cover a series of well known benchmark symbolic regression problems. Five synthetic problems were used: Keijzer-6 [13], Korns-12 [14], Vladislavleva-4 [36], Nguyen-7 [35], Pagie-1 [24]; and one real-world problem. All of them are suggestions made in [39], a recent survey on GP benchmarking.

Two performance measures are used to evaluate the different algorithms. First, fitness evaluation over the test set for the best solution found thus far. Second, the average population size, a relevant measure regarding the bloat phenomenon and solution complexity [34]. The evolution of these measures is analysed with respect to the total number of fitness function evaluations instead of generations, to account for the LS iterations. The LS performs a maximum of 400 iterations. However, we do not consider any additional computational effort due to the LS, which underestimates the computational cost of performing LS. The stopping criteria is the number of function evaluations. The GP configuration parameters for all variants are shown in Table 2. Some of these parameters vary depending on the problem, as suggested in [21]. For the case of Tower problem, total samples were divided in a ratio of 30/70% for the training and testing sets. As stated before, Table 1 summarizes the parameter values for the memetic GP variants.

6 Results and Summary

Figures 1-5 summarize the results of the all tested techniques, showing the median values computed over 30 independent runs. The first problem is Keijzer-6 shown in

[2] http://gplab.sourceforge.net/

Table 2. General GP parameters

Parameter	**Value**
Runs	30
Population	500
Function evaluations	2'000,000
Training set	as indicated in [21]
Testing set	as indicated in [21]
Crossover operator	Standard subtree crossover, 0.9 probability
Mutation operator	Mutation probability per node 0.05
Tree initialization	Ramped Half-and-Half, maximum depth 6
Function set	as indicated in [21]
Terminal set	Input variables, constants as indicated in [21]
Selection for reproduction	Tournament selection of size 3
Elitism	Best individual survives
Maximum tree depth	17

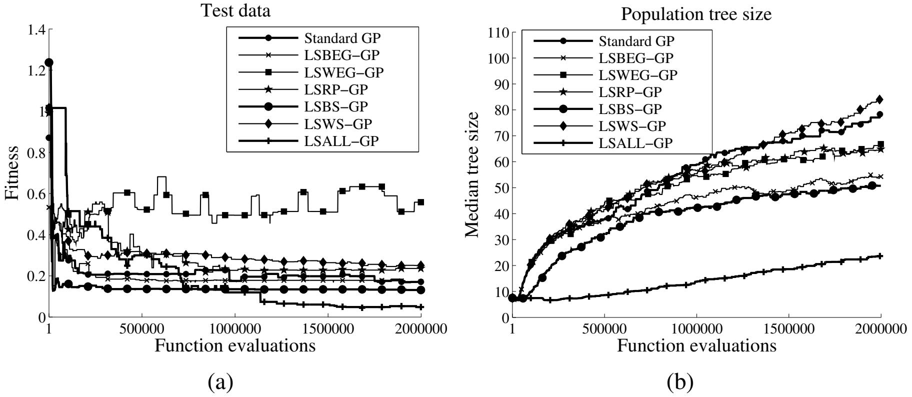


Fig. 1. Results for problem Keijzer-6 plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs

Figure 1, considered a simple or easy problem. The difficulty is raised with the testing data, since the solution must extrapolate outside the training domain. In general, most methods exhibit similar performance, with some notable exceptions. First, LSWEG-GP obviously achieves the worst test performance, while LSBS-GP and LSALL-GP exhibit the best. In particular, LSBS-GP and LSALL-GP converge quickly to very good performance; in fact LSBS-GP could have been halted very early in the run. However, after a million function evaluations LSALL-GP, shows a noticeable improvement, achieving the best results. In terms of size, both of these methods also exhibit the best results, with LSALL-GP producing the smaller trees.

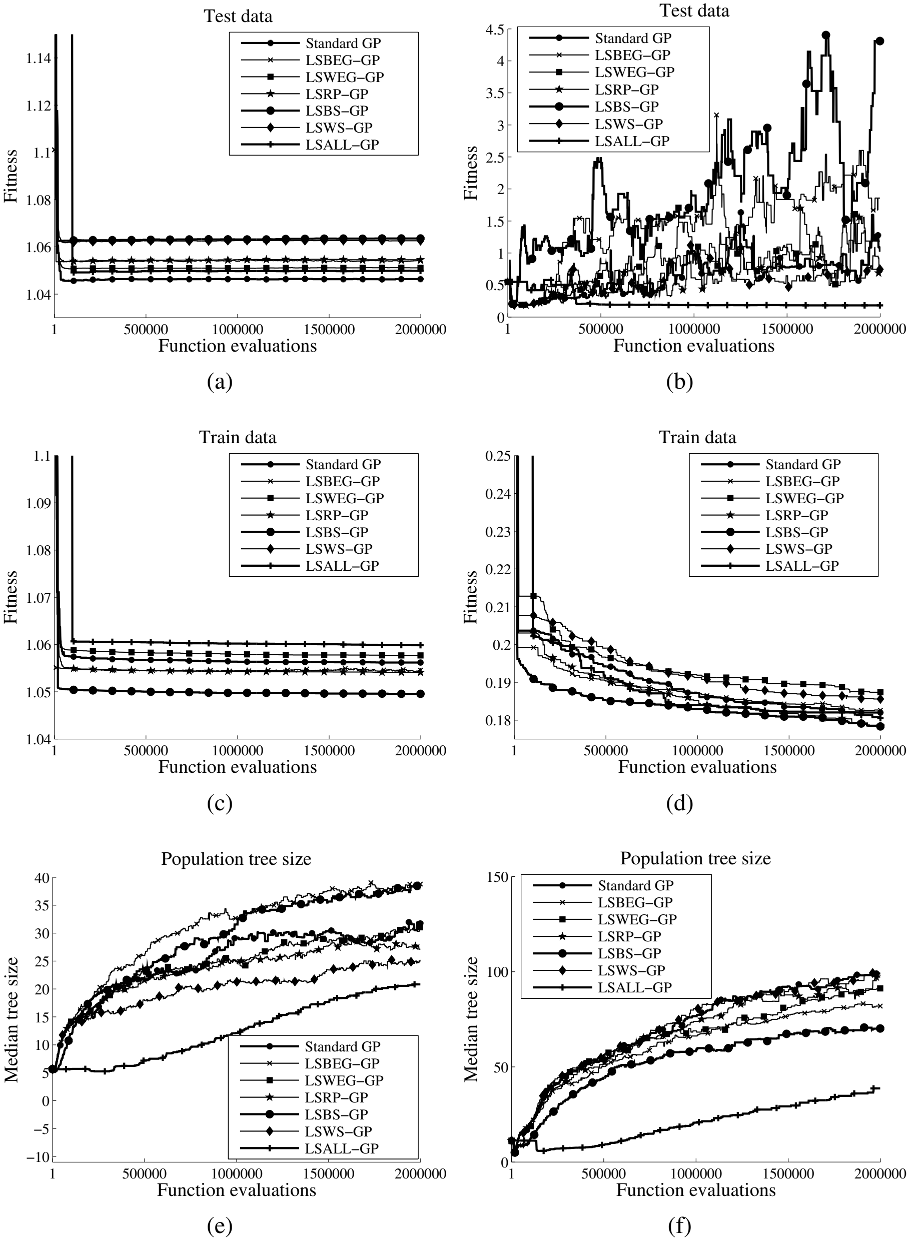


Fig. 2. Results for problems Korns-12 (a,c,e) and Vladislavleva-4 (b,d,f): (a,b) Fitness over test data; (c,d) Fitness over training data; and (e,f) Average program size. All plots show median values over 30 independent runs.

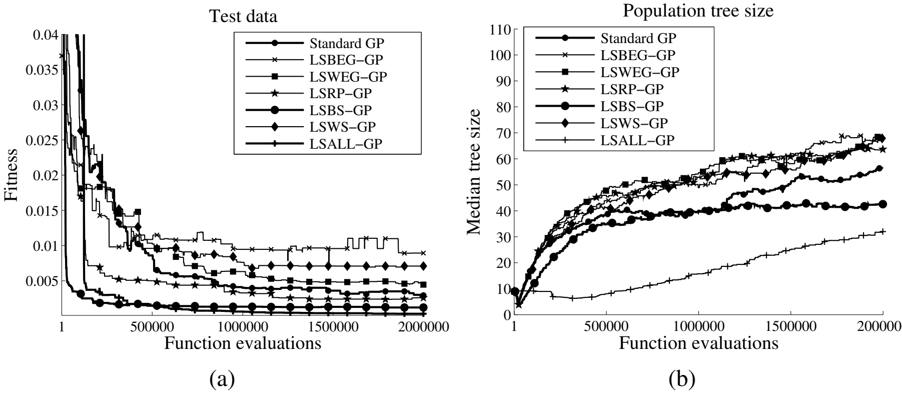


Fig. 3. Results for problem Nguyen-7 plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs.

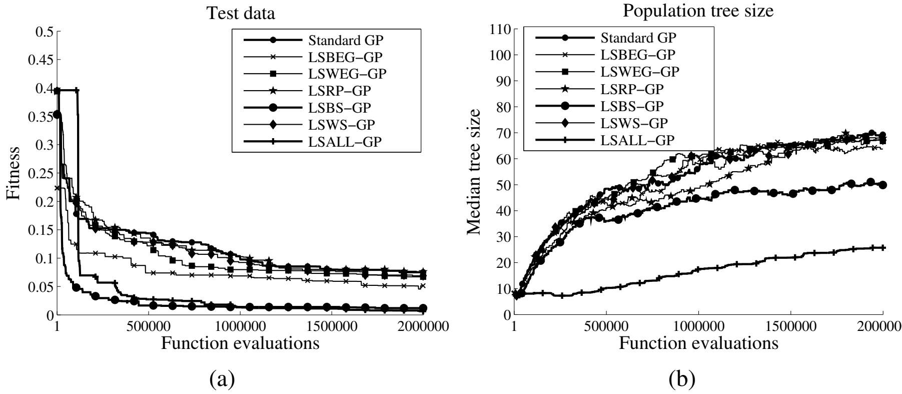


Fig. 4. Results for problem Pagie-1 plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs.

Figure 2 presents the results for the Korns-12 problem, considered difficult in GP literature since standard GP usually cannot find the true expression [14]. The problem includes redundant input data that does not influence its output. The idea is to test GP algorithms on their ability to avoid overfitting. With respect to test error, Figure 2(a) shows that most algorithms perform quite similarly, all of them converging to their best median values quite early in the runs. What is noticeable is the performance of LSBS-GP, with the worst test performance, but also the best training performance, shown in Figure 2(c), indicating a slight overfitting. Regarding the evolution of size, shown in Figure 2(e), all methods exhibit similar sizes, but code growth is noticeably slower for LSALL-GP.

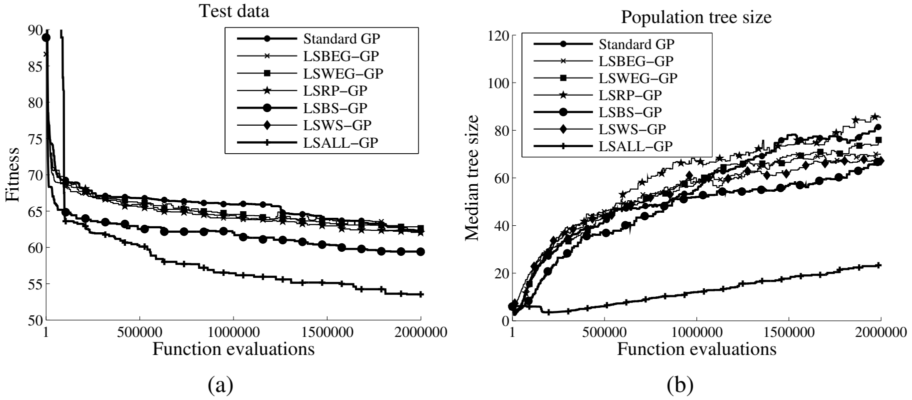


Fig. 5. Results for Tower problem plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs

Table 3. Summary of median fitness computed over the test set of each problem. The *Sample* column indicates the number of function evaluations performed; bold indicates the best results.

Problem	**Sample**	**S-GP**	**LSBEG-GP**	**LSWEG-GP**	**LSRP-GP**	**LSBS-GP**	**LSWS-GP**	**LSALL-GP**
Keijzer-6	250,000	0.2	0.2	0.5	0.54	**0.15**	0.29	0.46
	1000000	0.17	0.17	0.49	0.22	0.13	0.28	**0.11**
	2000000	0.17	0.17	0.5	0.23	0.13	0.25	**0.04**
Korns-12	250,000	**1.04**	1.05	1.05	1.05	1.06	1.06	**1.04**
	1000000	**1.04**	1.05	1.05	1.05	1.06	1.06	**1.04**
	2000000	**1.04**	1.05	1.05	1.05	1.06	1.06	**1.04**
Vladislavleva-4	250,000	**0.28**	0.47	0.31	0.35	1.03	0.34	0.5
	1000000	1.03	1.33	0.98	0.6	1.89	0.99	**0.18**
	2000000	1.01	1.85	1.18	0.68	4.3	0.74	**0.18**
Nguyen-7	250,000	0.02	0.01	0.02	0.005	**0.002**	0.02	0.003
	1000000	0.004	0.009	0.006	0.003	0.001	0.007	**0.0006**
	2000000	0.002	0.008	0.004	0.002	0.001	0.007	**0.0003**
Pagie-1	250,000	0.15	0.1	0.15	0.16	**0.02**	0.14	0.05
	1000000	0.1	0.06	0.07	0.1	0.01	0.09	**0.01**
	2000000	0.07	0.05	0.06	0.07	0.01	0.06	**0.006**
Tower	250,000	67.5	66.9	67	67.4	63.2	66.8	**61.98**
	1000000	65.92	64	64.54	64.39	62	64.52	**56.43**
	2000000	62.31	62.08	62.64	62.03	59.41	62.42	**53.53**

For the Vladislavleva-4 problem shown in Figure 2, considered a difficult test case for GP [36], overfitting is also noticeable. In Figure 2(b) we can see that test fitness varies greatly across the runs for most methods, particularly for LSBS-GP. Conversely in Figure 2(d) we can observe a smooth monotonic convergence for the training fitness. The case of LSBS-GP is noticeable, particularly since it is clear that the method achieves its best test fitness early in the run, and then starts to overfit. The only method that did not overfit was LSALL-GP, exhibiting the best performance over the test data. Finally, with respect to program size (Figure 2(f)), once again all methods show similar trends, with LSALL-GP producing substantially smaller trees.

Problem Nguyen-7 and Pagie-1 exhibit similar outcome in terms of fitness and program size. In both, among all methods LSBS-GP exhibits a faster convergence to a better solution quality computed over test data and also achieves the best results along with LSALL-GP. Once again LSALL-GP produces the smallest trees, with LSBS-GP the second best, however in these problems the difference between both of these methods is smaller than in the other cases. All other variants show similar performance based on both measures.

Finally, the Tower problem is an industrial real-world problem on modeling gas chromatography measurements of the composition of a distillation tower. This problem contains 5000 records and 23 potential input variables. The measurements (5000 for each variable) are not treated as time series, but simply used as samples for a regression model. In this case, LSALL-GP again achieves the most interesting results, both in regards to test-fitness and solution size. From the remaining methods, LSBS-GP shows the best performance, but still noticeably worse that LSALL-GP.

A final summary of the performance of each method is presented in Table 3, that shows three different snapshots of the median performance of each algorithm, after 250,000, 1 million and 2 million function evaluations. In general, LSALL-GP exhibits the best performance after all of the allowed function evaluations. However, if we only consider 250,000 function evaluations, then LSBS-GP exhibits the best performance on three of the six problems, and also the second best performance on two others.

From the remaining methods, those that show the worst performance are LSWEG-GP and LSWS-GP, that apply LS to the worst solutions in the population, what definitely seems to be a bad strategy. Moreover, while LSBS-GP shows good performance, the deterministic LSBEG-GP that applies LS to the best solution is notably inferior to the best methods, a noteworthy results since several previously published works have employed this memetic strategy.

7 Conclusions

This work studies the problem of integrating a local optimization process into a GP, using a Lamarckian HSE memetic approach. A comparative study is performed, evaluating different ways in which to incorporate a LS during the basic evolutionary process of GP, evaluating performance on symbolic regression problems. A simple tree parametrization is proposed, bounding the size of parameter space for each individual tree, even if bloat occurs during the run. The local optimization is done by a trust region technique that determines the optimal coefficients posing a basic non-linear curve fitting problem. As stated before, it is not clear what might be the best strategy to incorporate LS into GP, so different stochastic and deterministic variants are extensively evaluated over a set of widely used benchmark problems. In particular, each method uses different heuristic decisions to determine which individuals in the GP run should be subject to a local optimization process; experimental results suggest the following.

In general, it seems that a memetic GP almost always outperforms a standard GP, in terms of both solution quality and solution size. Moreover, among the different methods that were tested, several insights can be gathered. First, it does not appear to be beneficial to use a LS as another mutation strategy, such that all individuals might be candidates for a LS, given the average performance of LSRP-GP. Second, it does not seem useful to apply LS to the worst individuals in the population, as seen by the performance of LSWEG-GP and LSWS-GP. Third, many works have used the simple deterministic heuristic of applying LS to the best solution found, either at the end of the run or at each generation. However, this to does not seem to be an adequate strategy, given the performance of LSBEG-GP. Finally, of all the methods, the best performance was achieved when LS is applied to all of the solutions (LSALL-GP) or to random individuals chosen from the top percentile (w.r.t. fitness) of the population (LSBS-GP).

For all problems, the best performance was achieved by LSALL-GP. However, if only a small amount of computational effort is feasible, then LSBS-GP seems to be the best option, given its fast convergence. Moreover, the difference in computational effort between both methods is understated in the results presented here, since only the total iterations in the LS are considered, omitting the total effort devoted towards computing approximate derivatives or matrix inversions. Nevertheless, LSALL-GP also shows a substantial ability to curtail bloating during the search, this was indeed expected. Since the search process in LSALL-GP focuses primarily on parameter optimization, limiting the total of syntactic search performed by the GP crossover and mutation operators; i.e., the total number of generations is quite low for LSALL-GP, basically eliminating the possibility of bloating.

Future work will be centered on exploring and evaluating other parametrization schemes. Moreover, a method must be implemented that determines if a tree requires linear or non-linear parameter optimization, in order to simplify the process whenever possible and to use the parameter values as decision criteria to prune unnecessary subtrees. Finally, the GP search operators could be enhanced to explicitly take into account the parameter values of a tree.

Acknowledgments Funding provided by CONACYT (Mexico) Basic Science Research Project No. 178323, DGEST (Mexico) Research Projects No.5149.13-P and TIJ-ING-2012-110, and IRSES project ACoBSEC from the European Commission. First author supported by CONACYT (Mexico) scholarship No. 294213.

References

[1] Chen, X., Ong, Y.-S., Lim, M.-H., Tan, K.C.: A multi-facet survey on memetic computation. Trans. Evol. Comp. 15(5), 591–607 (2011)

[2] Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag New York, Inc., Secaucus (2006)

[3] Coleman, T.F., Li, Y.: On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds (1992)

[4] Coleman, T.F., Li, Y.: An interior trust region approach for nonlinear minimization subject to bounds. Technical report, Ithaca, NY, USA (1993)

[5] De Jong, K.: Evolutionary Computation: A Unified Approach. Bradford Book. Mit Press (2006)

[6] Dunn, E., Olague, G., Lutton, E.: Parisian camera placement for vision metrology. Pattern Recogn. Lett. 27(11), 1209–1219 (2006)

[7] Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2003)

[8] Emmerich, M., Grötzner, M., Schütz, M.: Design of graph-based evolutionary algorithms: A case study for chemical process networks. Evol. Comput. 9(3), 329–354 (2001)

[9] Eskridge, B., Hougen, D.: Imitating success: A memetic crossover operator for genetic programming. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation, June 20-23, pp. 809–815. IEEE Press, Portland (2004)

[10] Gill, P.E., Murray, W., Wright, M.H.: Practical optimization. Academic Press Inc. (Harcourt Brace Jovanovich Publishers), London (1981)

[11] Graff, M., Pea, R., Medina, A.: Wind speed forecasting using genetic programming. In: IEEE Congress on Evolutionary Computation, pp. 408–415. IEEE (2013)

[12] Hornby, G.S., Lohn, J.D., Linden, D.S.: Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission. Evol. Comput. 19(1), 1–23 (2011)

[13] Keijzer, M.: Improving symbolic regression with interval arithmetic and linear scaling. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 70–82. Springer, Heidelberg (2003)

[14] Korns, M.F.: Accuracy in symbolic regression. In: Riolo, R., Vladislavleva, E., Moore, J.H. (eds.) Genetic Programming Theory and Practice IX, May 12-14. Genetic and Evolutionary Computation, ch. 8, pp. 129–151. Springer, Ann Arbor (2011)

[15] Koza, J.: Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines 11(3), 251–284 (2010)

[16] Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge (1992)

[17] Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. Society for Industrial and Applied Mathematics (1995)

[18] Lohmann, R.: Application of Evolution Strategy in Parallel Populations. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 198–208. Springer, Heidelberg (1991)

[19] Luke, S.: Essentials of Metaheuristics, 2nd edn. Lulu (2013), `http://cs.gmu.edu/~sean/book/metaheuristics/`

[20] McConaghy, T.: FFX: Fast, scalable, deterministic symbolic regression technology. In: Riolo, R., Vladislavleva, E., Moore, J.H. (eds.) Genetic Programming Theory and Practice IX. Genetic and Evolutionary Computation, ch. 13, pp. 235–260. Springer, Ann Arbor (2011)

[21] McDermott, J., White, D.R., Luke, S., Manzoni, L., Castelli, M., Vanneschi, L., Jaskowski, W., Krawiec, K., Harper, R., De Jong, K., O'Reilly, U.-M.: Genetic programming needs better benchmarks. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, GECCO 2012, pp. 791–798. ACM, New York (2012)

[22] Moré, J.J., Sorensen, D.C.: Computing a trust region step. SIAM J. Scientific and Statistical Computing 4, 553–572 (1983)

[23] 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)

[24] Pagie, L., Hogeweg, P.: Evolutionary consequences of coevolving targets. Evolutionary Computation 5, 401–418 (1998)

[25] Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming. Lulu Enterprises, UK Ltd. (2008)

[26] Shultz, G., Schnabel, R., Byrd, R., Colorado Univ. at Boulder Dept of Computer Science: A Family of Trust Region Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties. Defense Technical Information Center (1982)

[27] Silva, S., Almeida, J.: Gplab–a genetic programming toolbox for matlab. In: Gregersen, L. (ed.) Proceedings of the Nordic MATLAB Conference, pp. 273–278 (2003)

[28] Silva, S., Costa, E.: Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. Genetic Programming and Evolvable Machines 10(2), 141–179 (2009)

[29] Smart, W., Zhang, M.: Continuously evolving programs in genetic programming using gradient descent. In: Mckay, R.I., Cho, S.-B. (eds.) Proceedings of The Second Asian-Pacific Workshop on Genetic Programming, p. 16. Cairns, Australia (2004)

[30] Sorensen, D.: Newton's Method with a Model Trust Region Modification. Defense Technical Information Center (1982)

[31] Spector, L.: Automatic Quantum Computer Programming: A Genetic Programming Approach (Genetic Programming). Springer-Verlag New York, Inc., Secaucus (2006)

[32] Steihaug, T.: The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. SIAM Journal on Numerical Analysis 20(3), 626–637 (1983)

[33] Topchy, A., Punch, W.F.: Faster genetic programming based on local gradient search of numeric leaf values. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), July 7-11, pp. 155–162. Morgan Kaufmann (2001)

[34] Trujillo, L., Naredo, E., Martínez, Y.: Preliminary study of bloat in genetic programming with behavior-based search. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 293–305. Springer, Heidelberg (2013)

[35] Uy, N.Q., Hoai, N.X., O'Neill, M., Mckay, R.I., Galván-López, E.: Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Genetic Programming and Evolvable Machines 12(2), 91–119 (2011)

[36] Vladislavleva, E.J., Smits, G.F., Den Hertog, D.: Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. Trans. Evol. Comp. 13(2), 333–349 (2009)

[37] Wagner, S., Kronberger, G.: Algorithm and experiment design with heuristic lab: An open source optimization environment for research and education. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion, GECCO Companion 2012, pp. 1287–1316. ACM, New York (2012)

[38] Wang, P., Tang, K., Tsang, E.P.K., Yao, X.: A memetic genetic programming with decision tree-based local search for classification problems. In: IEEE Congress on Evolutionary Computation, pp. 917–924. IEEE (2011)

[39] White, D.R., McDermott, J., Castelli, M., Manzoni, L., Goldman, B., Kronberger, G., Ja'skowski, W., O'Reilly, U.-M., Luke, S.: Better gp benchmarks: community survey results and proposals. Genetic Programming and Evolvable Machines 14(1), 3–29 (2013)

[40] Worm, T., Chiu, K.: Prioritized grammar enumeration: Symbolic regression by dynamic programming. In: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference, GECCO 2013, pp. 1021–1028. ACM, New York (2013)

[41] Yuan, J.Y.: Numerical methods for generalized least squares problems. Journal of Computational and Applied Mathematics 66(1-2), 571–584 (1996)

[42] Zhang, M., Smart, W.: Genetic programming with gradient descent search for multiclass object classification. In: Keijzer, M., O'Reilly, U.-M., Lucas, S., Costa, E., Soule, T. (eds.) EuroGP 2004. LNCS, vol. 3003, pp. 399–408. Springer, Heidelberg (2004)

Part VII

Evolutionary Multiobjective Optimization

Application of the MOAA for the Optimization of Wireless Sensor Networks

Valerio Lattarulo and Geoffrey T. Parks

University of Cambridge
Department of Engineering
Engineering Design Centre
Cambridge, Trumpington Street, CB2 1PZ, UK
{vl261,gtp10}@cam.ac.uk

Abstract. The multi-objective alliance algorithm (MOAA), a recently introduced optimization algorithm, is applied to the optimization of wireless sensor network layouts. Two different networks with 10 and 50 sensors respectively are optimized. MOAA performance is compared with that of NSGA-II and SPEA2 for 1000, 2000, 3000 and 5000 function evaluations for both networks. The epsilon and hypervolume indicators and the Kruskal-Wallis statistical test are used for the performance comparison. The results show that in most cases the MOAA outperforms both NSGA-II and SPEA2 on both versions of this problem.

1 Introduction

Wireless sensor networks (WSN) are networks of autonomous nodes linked to each other and distributed in an environment. Generally, the nodes contain several sensors, such as temperature and pressure sensors, in order to provide useful information about environments that are not physically reachable or that need continuous monitoring. These networks are applied in many different fields such as military (enemy detection), natural phenomena (e.g. forest fire or earthquake detection) and data logging [1].

WSN are typically composed of many nodes and sensors and finding a good layout that guarantees a certain coverage and system lifetime is a complex real-world optimization problem. Deterministic approaches are not particularly suitable for this typology of problems, because many variables are involved and there are many local minima and non-linearities. One possible way to overcome these problems is by using multi-objective (MO) metaheuristic approaches. These methods have already proven to be more successful than deterministic gradient-based methods for complex MO problems in many different fields. Some of the best-known algorithms in the state of the art are based on Genetic Algorithms [2] such as NSGA-II [3], SPEA2 [4] and MOEA/D [5]. These methods are generally preferred because they use a population that can be naturally tuned to solve MO problems by finding several Pareto-optimal (PO) solutions in parallel; thus MO evolutionary approaches are the most widely used [6, 7]. Various methodologies, based on different metaheuristic approaches, have already been used to tackle several variants of WSN problems: to maximize lifetime and coverage [8], to maximize coverage and minimize the number of sensors used [9], to maximize the coverage of large networks [10]. The authors of [8] use the algorithm ‘out of the box’ but

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_16, © Springer International Publishing Switzerland 2014

in other studies, such as [11] and [12], problem-specific operators/techniques which enhance algorithm performance are developed. Such developments can help avoid unnecessary searches and function evaluations but diminish the generality of the algorithm.

The Multi-objective Alliance Algorithm (MOAA) [13] is an evolutionary approach based on the Alliance Algorithm (AA) which is a relatively new single-objective optimization algorithm that has been applied successfully to many different typologies of problems [14–17]. The first MO version was compared with NSGA-II [3] and SPEA2 [4]. That study [13] revealed a certain complementarity because the three approaches offered superior performance for different classes of problems. Since then, a mixed-integer version of the MOAA with hybrid components has been developed. This was able to outperfom a hybrid version of NSGA-II on a satellite constellation refueling optimization problem [18]. The knowledge acquired in solving benchmark and complex real-world problems led to the development of a new version of the MOAA which has already been tested on MO benchmark problems [19], the optimization of a supersonic airfoil [20] and a benchmark aerodynamic optimization problem [21].

In this paper, the MOAA is used ‘out of the box’ and applied to the optimization of WSN layouts to test whether the approach is able to perform well without needing to tune its parameters or modify the algorithm.

The main aims for these WSN problems are to maximize the coverage and the lifetime of the network given a certain number of sensors. The results obtained using the MOAA are compared with those of NSGA-II and SPEA2.

The remainder of this paper is structured as follows: Sect. 2 introduces the WSN problem; Sect. 3 explains the optimization framework; Sect. 4 provides an overview of how the MOAA works; Sect. 5 introduces the indicators and statistical tests used for the evaluation of the algorithms; Sect. 6 reports on the MOAA’s performance on these problems, comparing it with those of NSGA-II and SPEA2; Sect. 7 discusses the overall performance of the algorithms and describes several layouts created by the algorithms; Sect. 8 concludes the paper and proposes possible future work.

2 Wireless Sensor Networks

A typical WSN is composed of a high-energy communication node (HECN) base and many nodes: the HECN receives information from all the nodes distributed in the environment. Each node in the WSN is essentially comprised of four parts: a microcontroller to execute the calculations, a battery to provide energy to the node, an antenna to send and receive data, and some sensors to gather the necessary infomation from the environment. The nodes can send information within a certain distance R_{com} and can use the sensors within a radius R_{sens}. In this study, without loss of generality, R_{com} and R_{sens} have the same value. Fig. 1 shows the HECN in the centre with radius R_{com} (blue circle) and two nodes. One of the nodes is connected to the HECN (black circle) but the other is not (red circle) because it is too far away (the distance exceeds R_{com}).

Every node can be directly connected to the HECN or can send data via another node that is connected to it. Thus, the nodes can send their own data and that of other nodes: this ability enhances the coverage of the environment but reduces the overall WSN lifetime because some nodes consume more energy than others by sending more data.

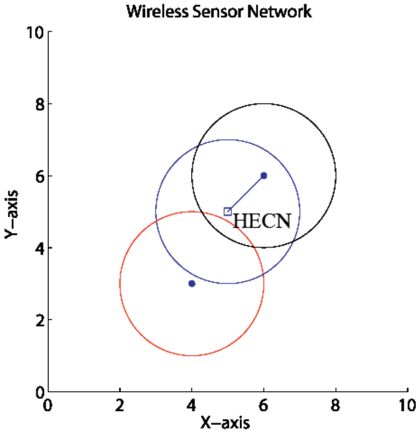


Fig. 1. The main actors in a WSN

The quality of a WSN is defined by several factors (costs, data speed, etc.), but two of the most important are the coverage and lifetime of the network. These two factors are in conflict because at one extreme all the nodes are directly connected to the HECN but they cannot cover all the environment, while at the other extreme the nodes can create several branches in order to increase the coverage but the lifetime is adversely affected.

3 Optimization Framework

In the model, developed in Matlab, the targeted area A is a 2D plane $L \times L$, where L is the side length of the plane. The WSN is composed of n nodes. Each node has a position defined by two variables x and y (respectively the the x and y coordinates). Thus, the number of variables in the problem is $2n$.

It is important to make clear that not all the nodes need to be effectively used: the algorithm can connect up to 10 nodes in the first test and up to 50 in the second. Any nodes that are not connected in any way to the central node (the one connected to the HECN) are not considered in the calculation of the coverage and lifetime of the network.

$A_{cov}(x_i, y_i, R_{sens})$ is the function that calculates the area covered by a given node i. The coverage objective is intuitively defined as the union of the areas covered by all the nodes divided by the total area (Eq. 1):

$$Coverage = \frac{\bigcup_{i=1}^{n} A_{cov}(x_i, y_i, R_{sens})}{A} \tag{1}$$

In practice, instead of calculating the areas covered by all the nodes, the coverage is calculated by creating a grid of G uniformly distributed points that cover all the targeted area. The probability that a point p with a position (x_p, y_p) is covered by node i is given by:

$$P_{node}(p, i) = \begin{cases} 1 & \text{if } (x_p - x_i)^2 + (y_p - y_i)^2 \leq R_{sens} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Hence the probability that a point p is covered by the WSN can be found as:

$$P_{cov}(p) = H\left(\left(\sum_{i=1}^{n} P_{node}(p,i)\right) - 1\right), \tag{3}$$

where H is the Heavyside step function.

The total coverage is then estimated by summing the coverage probabilities for all G points and dividing by the total number of points:

$$Coverage_{Approx} = \frac{\sum_{i=1}^{G} P_{cov}(i)}{G} \tag{4}$$

This approach finds an approximation of the real coverage but, with sufficient points, the result is accurate enough for the purposes of this paper.

The lifetime objective is defined as the ratio of the time that the first node fails due to energy shortage, T_{fail}, and the maximum lifetime of the node, T_{max}:

$$Lifetime = \frac{T_{fail}}{T_{max}} \tag{5}$$

It is assumed that the nodes are synchronized. At every cycle the nodes need to transmit their information to the HECN. If they are not directly connected to the HECN, they need to use the shortest route that maximizes the remaining energy of the network. This is found using the Dijkstra algorithm [22] with the inverse of the remaining energy of the nodes as weights. Every time a node sends some data the weights are updated. The initial energy T_{max} has a normalized value of 100 and the energy is decreased by 1 at each data-sending operation. The operation is repeated until a node fails due to energy shortage.

In these problems $L = 10$ and, without loss of generality, the HECN is at the centre of the area (position $[5, 5]$). The step size between the grid points is 0.1 (a fine grid in order to reduce the error in estimating the coverage) giving a total of $101 \times 101 = 10201$ points. Two networks with different numbers of sensors are considered: the first has 10 nodes (20 variables) and $R_{sens} = 2$; the second has 50 nodes (100 variables) and $R_{sens} = 1$. Each sensor can be placed anywhere in the targeted area. Thus, every variable can assume any real value from 0 to 10. The two objectives are the minimization of $-Coverage_{Approx}$ and $-Lifetime$.

4 Multi-Objective Alliance Algorithm

The MOAA is a metaheuristic optimization algorithm inspired by the metaphorical idea of a number of tribes struggling to conquer an environment that offers resources that enable them to survive. The tribes are characterized by two features: the skills and resources necessary for survival. Tribes try to improve skills by forming alliances, which are also characterized by the skills and resources needed, but these now depend on the tribes within the alliance. The two main search elements of the algorithm are the formation of alliances and the creation of new tribes. One MOAA cycle ends when the strongest possible alliances of existing tribes have been created. The algorithm then begins a new cycle starting with new tribes whose creation is influenced by the previous strongest alliances.

In the general form of the MOAA, a tribe t is a tuple (x_t, s_t, r_t, a_t) composed of: a point in the solution space x_t; a set of skills dependent on the values of the N_S objectives evaluated at x_t; a set of resource demands dependent on the values of the N_R constraint functions (in this case there are no constraints); an alliance vector a_t containing the IDs of the tribes allied to tribe t.

An alliance is a mutually disjoint partition of tribes. Each alliance a forms a new point x_a in the solution space defined by the tribes in the alliance. The sets of skills s_a and resource demands r_a of the alliance consist of the objective and constraint functions S and R evaluated at x_a.

4.1 Algorithm Steps

The procedure followed by the MOAA can be divided into several steps. This version of the algorithm has been already described in detail in other papers. For this reason, only a general description (without equations) of the steps is provided here. A detailed description of these steps can be found in [20]. A general definition of the framework is provided in [23], a copy of which is available from the first author on request.

Solution Generation. In the MOAA's first cycle the tribes (solutions) are chosen randomly (with a uniform distribution). In subsequent cycles: some tribes are copies of previously found PO solutions; others are modifications (using a normal distribution with an adaptive standard deviation σ) of PO solutions.

An important feature is the adaptive nature of σ: this parameter adaptively decreases in order to produce high diversity at the start of the optimization and low diversity at the end. This mechanism enhances the initial exploration of the solution space and the final convergence of the solutions already found.

Verification. In this phase an alliance/tribe (A/T) tries to forge an alliance (a point in solution space x_a) with another tribe. When the alliance is created, x_a is made up of components drawn from the tribes within the alliance plus some variation. Thus, x_a is created by using uniform recombination between all the tribes of the alliance with variation applied randomly to some of the variables.

The standard deviation for the variation depends on the difference between the highest and lowest values of the corresponding variable among the tribes within the alliance; the variation for an alliance of tribes that are close together is small (local search) and for far-apart tribes it is large (global search). Generally at the start of an optimization the tribes within an alliance are far apart and then they start to come closer together. This behavior can be viewed as an initial global search followed by progressively more localized search.

The new alliance will only be confirmed if at least one skill in s_a of x_a is better than one skill in s_{t_1} of the solution representing the A/T seeking to forge the alliance and one skill in s_{t_2} of the tribe chosen to become an ally.

Alliance and Data Structure Update. There are two possible outcomes from the Verification Phase: the chosen tribe joins the A/T, forming a new alliance, or the tribe does not join and the new alliance is not confirmed. Next there is an update of the data structures necessary for the low level system to function (all the tribes need to be informed of any change in the environment). The cycle termination conditions are also checked. The cycle finishes when each A/T has tried to form a new alliance with every other tribe and remains unchanged. If this condition is not met, the algorithm continues to try to form new alliances.

Selection of the Strongest Alliances and Termination. At the end of the interactions between tribes, many alliances will have been formed but only the strongest A/Ts will conquer the environment. Therefore the A/Ts selected are the non-dominated points in objective space. These correspond to the best solutions to the problem found thus far. They can be used as the input to another MOAA cycle or, if the algorithm has ended, they represent the final results.

There is a limit N_{tot} to the number of best solutions saved in the archive of PO solutions. If the number of non-dominated solutions exceeds this, then all the solutions with at least one neighbor within a neighborhood distance d (in objective space) are eliminated. The initial value of d is 0 and then changes adaptively. This formulation recognizes that d should depend on: the previous value of d, the current number of solutions, the number of function evaluations that can be afforded, and the actual ranges of the Pareto front.

The MOAA is terminated when a specified limit E_{tot} on the number of solution evaluations is reached. The algorithm output is then the best solutions and the Pareto front found.

Extended Archive. The MOAA also uses an Extended Archive which saves some dominated solutions that could help maintain diversity among solutions and the convergence of the algorithm. It accomplishes this task essentially by finding every large gap in the Pareto front (larger than the average gap between solutions multiplied by a factor d_f) in all the dimensions of the objective space and saving all the non-dominated solutions (and solutions that nearly satisfy the constraints) found in these gaps. The factor d_f changes adaptively, increasing over time, as the gaps between the solutions become smaller, reaching similar values: by increasing its value only large gaps (in comparison with the average) are taken into consideration.

5 Indicators and Statistical Test

The performance measures chosen to evaluate the algorithms are the *epsilon* and *hypervolume* indicators provided in the PISA package [24]. The epsilon indicator [25] makes use of the Pareto-dominance concept and measures, given a reference set of points (ideally the true Pareto front, if available), the minimum amount ϵ necessary to translate all the points of the found Pareto front to weakly dominate the reference set. The hypervolume indicator [26] calculates the difference between the hypervolume of the space dominated by the found Pareto front and the hypervolume of the space dominated by a reference set (again, ideally the true Pareto front). This indicator needs a reference point which is dominated by all the found points in order to bound the hypervolume.

The statistical test chosen for result evaluation is the Kruskal-Wallis test, provided in the PISA package [24]. This is a non-parametric rank-based test that can be used to compare two independent sets of sampled data. It outputs p-values that estimate the probability of rejecting the null hypothesis of the study question when that hypothesis is true. Here the p-values can be interpreted as the probability that the MOAA is superior to NSGA-II or SPEA2 only by chance.

6 Results

The algorithms were tested on two different networks (NET1 and NET2). NET1 has 10 sensors and NET2 has 50. In both cases four tests were performed with 1000, 2000, 3000 and 5000 solution evaluations and each test was repeated 20 times. The NSGA-II implementation used is a Matlab version [27] and the parameters used are specified in [3]. The SPEA2 implementation used is also a Matlab version [28] and the parameters used are specified in [29]. The MOAA parameters used are shown in Table 1 where N_{vars} is the number of variables in the problem (20 for NET1 and 100 for NET2). These parameters have already proven their effectiveness in other studies with standard benchmark functions [19] such as the ZDT [30] and DTLZ [31] families and also with real-world optimization problems [20, 21].

Table 1. MOAA Parameters

Parameter	Value	Description
N	6	Number of tribes
P_1	0.5	Probability 1 for the creation of tribes
P_2	0.2	Probability 2 for the creation of tribes
σ_{init}	0.3	Initial standard deviation
σ_{end}	0.01	Final standard deviation
P_3	$2/N_{vars}$	Probability for the creation of alliances
σ_a	0.1	Standard deviation for the creation of alliances
N_{tot}	100	Total number of PO solutions
N_f	10	Factor for evaluation neighborhood

The PO solutions found in each of the 20 runs for all four tests on NET1 are shown in Fig. 2 on the left, while the PO solutions amongst all 20 runs are shown in Fig. 2 on the right. The graphs show that the Pareto front is discrete: with a small number of sensors it is possible to create a limited number of layouts. In all the tests the MOAA performs best, followed by SPEA2 and then NSGA-II. The MOAA is able to find a wider Pareto front and the right side of Fig. 2 shows that there is a gap between the MOAA Pareto front and the other Pareto fronts.

Results from the 20 runs were used to compute the means and standard deviations of the epsilon and hypervolume indicators and the corresponding p-values. Lower values

Table 2. NET1: Comparison with Epsilon Indicator

E_{tot}	MOAA		NSGA-II		SPEA2	
	Mean	Std	Mean	Std	Mean	Std
1000	0.392	0.073	0.490	0.065	0.483	0.057
2000	0.301	0.080	0.392	0.074	0.392	0.073
3000	0.239	0.069	0.327	0.069	0.329	0.052
5000	0.191	0.059	0.272	0.058	0.261	0.050

Table 3. NET1: Comparison with Hypervolume Indicator

E_{tot}	MOAA		NSGA-II		SPEA2	
	Mean	Std	Mean	Std	Mean	Std
1000	0.301	0.047	0.419	0.038	0.411	0.038
2000	0.219	0.049	0.336	0.044	0.303	0.039
3000	0.166	0.051	0.289	0.038	0.239	0.034
5000	0.119	0.044	0.233	0.036	0.173	0.025

Table 4. NET1: Kruskal-Wallis Statistical Test

E_{tot}	p_N		p_S	
	Eps	Hyp	Eps	Hyp
1000	$2.7 \cdot 10^{-5}$	$5.4 \cdot 10^{-11}$	$5.0 \cdot 10^{-5}$	$1.8 \cdot 10^{-10}$
2000	$1.5 \cdot 10^{-4}$	$7.8 \cdot 10^{-10}$	$1.0 \cdot 10^{-4}$	$2.0 \cdot 10^{-8}$
3000	$8.9 \cdot 10^{-5}$	$3.8 \cdot 10^{-10}$	$2.3 \cdot 10^{-5}$	$1.8 \cdot 10^{-7}$
5000	$3.7 \cdot 10^{-5}$	$8.2 \cdot 10^{-11}$	$3.2 \cdot 10^{-5}$	$8.6 \cdot 10^{-7}$

of these metrics indicate better performance. The reference set for these comparisons comprised the PO solutions found by all the algorithms.

Table 2 shows the mean and standard deviation of the epsilon indicator. The MOAA has the best mean in all four tests, while NSGA-II and SPEA2 obtain similar values. These values confirm the observations based on Fig. 2. The standard deviations of the three approaches are similar but the MOAA yields slightly higher values.

Hypervolume indicator metrics are shown in Table 3. The MOAA again obtains the best mean in all four tests, followed by SPEA2 and NSGA-II. After 5000 solution evaluations the MOAA mean is approximately half that of NSGA-II. Again, the MOAA standard deviation is slightly higher.

To quantify the efficacy of these two metrics in characterizing the algorithms' relative success, we compute the p-values obtained with the Kruskal-Wallis statistical test. Lower p-values indicate that one algorithm is better than the other. These values are shown in Table 4. Here we compute the probability that the MOAA provides a better set of PO solutions *by chance*. p_N values are the p-values for NSGA-II; p_S values are the p-values for SPEA2. These results confirm previous observations. All the p-values are very close to 0 (especially so for the hypervolume indicator). This means that the MOAA performs consistently better on the NET1 problem than both NSGA-II and SPEA2.

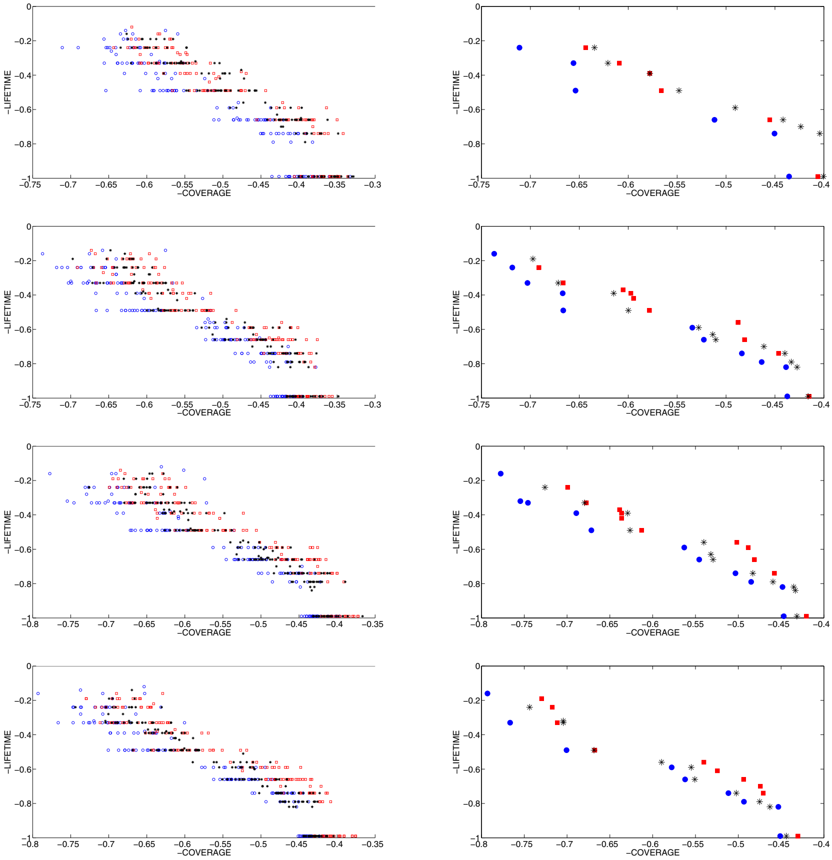


Fig. 2. NET1: Comparison of the results from 20 runs on the left and comparison of the PO solutions found on the right. Blue circles are MOAA solutions, red squares NSGA-II solutions, black asterisks SPEA2 solutions. From top to bottom: 1000, 2000, 3000, 5000 evaluations.

The same procedure is repeated for NET2. Fig. 3 shows the PO solutions found in each of the 20 runs for all four tests on the left and the PO solutions amongst all the runs on the right. In this case the Pareto front is less discretized because with 50 sensors it is possible to create more layouts. It is again noticeable that the MOAA's solutions represent superior Pareto fronts, followed by SPEA2 and then NSGA-II. In this case, SPEA2 is able to find a wider front, but only a few solutions lie in this extended part. In particular, SPEA2 is able to find a solution with a coverage of 0.52 but it is the only one of its kind. Unfortunately, such solutions do not have practical utility in many common WSN applications because the associated network lifetime is less than 5–10%.

The right side of Fig. 3 shows that the MOAA always finds the best Pareto front overall and its solutions are reasonably well spaced in comparison with the other two algorithms'.

Table 5. NET2: Comparison with Epsilon Indicator

E_{tot}	MOAA		NSGA-II		SPEA2	
	Mean	Std	Mean	Std	Mean	Std
1000	0.489	0.070	0.526	0.086	0.444	0.092
2000	0.389	0.058	0.485	0.077	0.335	0.073
3000	0.333	0.063	0.438	0.076	0.274	0.062
5000	0.291	0.066	0.383	0.092	0.212	0.069

Table 6. NET2: Comparison with Hypervolume Indicator

E_{tot}	MOAA		NSGA-II		SPEA2	
	Mean	Std	Mean	Std	Mean	Std
1000	0.223	0.023	0.266	0.022	0.240	0.015
2000	0.166	0.015	0.234	0.026	0.183	0.018
3000	0.134	0.018	0.208	0.024	0.152	0.015
5000	0.094	0.017	0.178	0.026	0.112	0.021

Table 7. NET2: Kruskal-Wallis Statistical Test

E_{tot}	p_N		p_S	
	Eps	Hyp	Eps	Hyp
1000	$H0$	$4.0 \cdot 10^{-8}$	$H0$	$2.4 \cdot 10^{-3}$
2000	$1.4 \cdot 10^{-5}$	$2.3 \cdot 10^{-11}$	$H0$	$1.3 \cdot 10^{-3}$
3000	$6.0 \cdot 10^{-6}$	$7.9 \cdot 10^{-13}$	0.99	$1.3 \cdot 10^{-3}$
5000	$2.9 \cdot 10^{-4}$	$2.6 \cdot 10^{-13}$	1	$3.5 \cdot 10^{-3}$

Table 5 shows the mean and standard deviation of the epsilon indicator. In this case, SPEA2 has the best mean in all four tests, followed by the MOAA and NSGA-II. SPEA2 obtains better values because this indicator is quite sensitive to the edges of the Pareto front and in a few cases SPEA2 was able to find a wider Pareto front. The MOAA yields the lowest standard deviation values.

Hypervolume indicator metrics are shown in Table 6. In this case, the MOAA obtains the best mean in all four tests, followed by SPEA2 and NSGA-II. The MOAA's better results are due to the fact that the hypervolume indicator is more sensitive to the disposition of the solutions in the Pareto front and the MOAA's solutions are better spaced than those found by the other two algorithms.

The p-values from the Kruskal-Wallis test are shown in Table 7. In this case, the MOAA is always able to outperform the other two algorithms, according to the hypervolume indicator. On the other hand, according to the epsilon indicator, there are three different cases: the three algorithms yield similar performance for 1000 solution evaluations; the MOAA and SPEA2 yield similar perfomance and are better than NSGA-II for 2000 solution evaluations; the MOAA is better than NSGA-II but worse than SPEA2 for the last two cases. These results confirm previous observations. For NET2 the MOAA is a sensible choice: it yields better or equally good results 14 out of 16 times (87.5%) in the Kruskal-Wallis test.

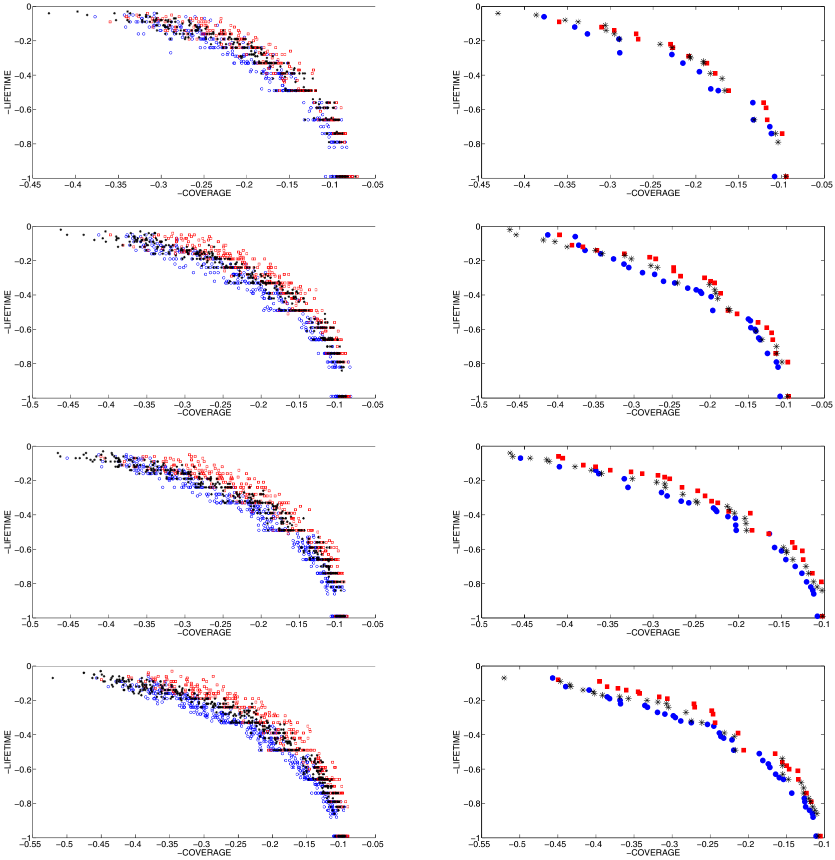


Fig. 3. NET2: Comparison of the results from 20 runs on the left and comparison of the PO solutions found on the right. Blue circles are MOAA solutions, red squares NSGA-II solutions, black asterisks SPEA2 solutions. From top to bottom: 1000, 2000, 3000, 5000 evaluations.

7 Discussion

The cases with 10 and 50 sensors confirm that the MOAA is able to yield good performance on such WSN problems without the need to tune its parameters or modify the algorithm. The MOAA was able to outperform two state-of-the-art algorithms in most of the tests. These results confirm the good performance of the MOAA that has already been observed on benchmark functions [19] and with real-world optimization problems [20, 21].

For NET1, the MOAA gave the best performance in every test. For NET2, it gave the best performance in the first two tests and similar performance to that obtained by SPEA2 after 3000 solution evaluations, because, although the MOAA finds Pareto fronts with better convergence and diversity, in some cases SPEA2 finds wider Pareto

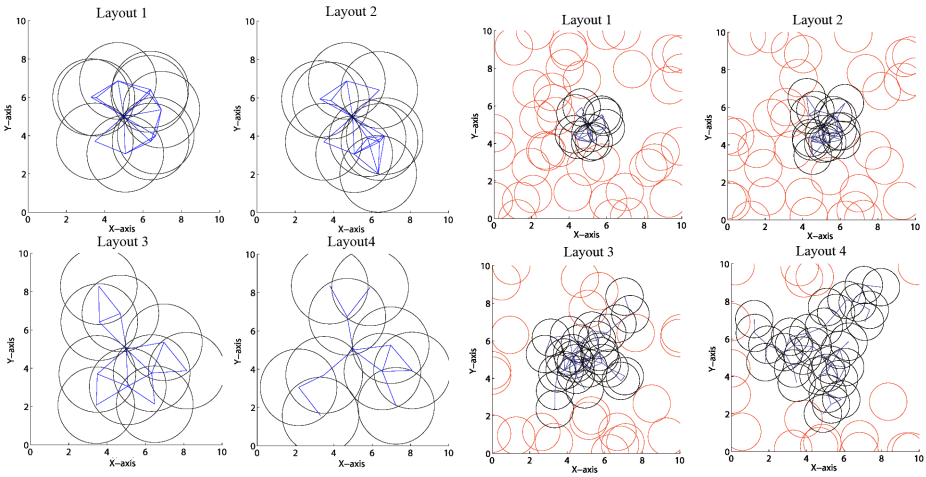


Fig. 4. On the left: Examples of four main layouts for NET1. On the right: Examples of four main layouts for NET2. Red circles denote unused sensors.

fronts. This leads to two different outcomes: SPEA2's performance is better according to the epsilon indicator, the MOAA's performance is better according to the hypervolume indicator. The additional solutions found by SPEA2 at the edge of the Pareto front cannot be used in practice, however, due to the low network lifetime; in contrast, the well-spaced solutions at the 'knee' of the Pareto front found by the MOAA are effective practical solutions.

The Pareto front solutions correspond to different sensor layouts. It is interesting to visualise the differences in layouts. Fig. 4 shows four main layout types for NET1 and NET2. These come from MOAA Pareto fronts, but all three algorithms found the same layout types with slight differences in sensor positions.

In the left side of Fig. 4, Layout 1 has all the sensors directly connected to the HECN, forming a characteristic 'flower' shape. This solution is at one edge of the Pareto front: all solutions with this layout have maximum lifetime values (of unity) and the coverage value is low (0.45). In Layout 2 some sensors are not directly connected to the HECN. These sensors need to send data through other sensors to the HECN. This affects the network lifetime (in this case it is 0.8) but improves the coverage (to 0.49). Layout 3 is formed by sensors that are divided into several branches. Each branch is connected to just two or three sensors in order to maximize the coverage. The general shape is less chaotic and the sensors are well-spaced. In this case the lifetime is 0.56 and the coverage is 0.62. In Layout 4 there are sensors connected to only one sensor. This is the minimum possible connection. It maximizes the coverage but greatly reduces the network lifetime. In this example, the lifetime is 0.33 and the coverage is 0.71.

In all the NET1 layouts there are no unused sensors. This is not true for NET2 where it is possible to find several unused sensors because it is more difficult to use all 50 sensors in each layout. The MOAA was, in general, able to use more sensors than SPEA2 or NSGA-II, resulting in better Pareto fronts.

In the right side of Fig. 4, Layout 1 again has all the sensors directly connected to the HECN forming a 'flower'. This solution has a lifetime of 1, the coverage is 0.11 and there are 40 unused sensors. Layout 2 has some sensors that are not directly connected to the HECN. Its lifetime is 0.61, the coverage is 0.16 and 35 sensors are unused. Layout 3 again contains several branches. The lifetime is now 0.24, the coverage 0.30 and 23 sensors are unused. In Layout 4 many sensors are connected to just one other sensor. Its lifetime is 0.04, the coverage is 0.42 and 18 sensors are unused. It is clear that the more sensors that are used the higher the coverage but the lower the network lifetime – a strong trade-off. None of the algorithms were able to find layouts using all the sensors. This means that the Pareto fronts are not optimal: longer runs are needed to find better solutions.

8 Conclusions

In this paper, the MOAA was applied to the optimization of wireless sensor networks in order to maximize their coverage and lifetime. Two different instances were tackled and the results were compared with those given by NSGA-II and SPEA2. The following are the key conclusions of this study:

1. Overall, the MOAA outperformed NSGA-II and SPEA2, finding better Pareto fronts. This is confirmed by the p-values for the epsilon and hypervolume indicators versus NSGA-II and by the hypervolume indicator versus SPEA2.
2. Pareto fronts found by the MOAA are typically better distributed and contain more solutions. SPEA2-found fronts are better in these regards than those of NSGA-II.
3. The use of the MOAA 'out of the box', without performing any modification or parameter tuning, has demonstrated its utility as a general problem solver.
4. The MOAA has shown its competitiveness with other state-of-the-art algorithms on a problem with many variables (NET2).

In future work, the MOAA will be applied to other networks with increased numbers of sensors and function evaluations to investigate its longer term performance. These networks will be also tested with different values of R_{sens} and R_{com} to investigate how these affect the layouts. The number of unused sensors will be included as a third objective in problems to guarantee a minimum number of sensors for each type of layout. MOAA performance will also be compared with that of other algorithms that have proved successful on WSN problems.

References

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. 38(4), 393–422 (2002)

[2] Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge (1992)

[3] 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)

[4] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. In: Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100 (2001)

[5] Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 11(6), 712–731 (2007)

[6] Coello Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation. Springer, Secaucus (2006)

[7] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms, 1st edn. Wiley, Chichester (2001)

[8] Jourdan, D., de Weck, O.L.: Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. In: IEEE Semiannual Vehicular Technology Conference, Milan, Italy, pp. 2466–2470 (2004)

[9] Jia, J., Chen, J., Chang, G., Li, J., Jia, Y.: Coverage optimization based on improved NSGA-II in wireless sensor network. In: IEEE International Conference on Integration Technology, Shenzhen, China, pp. 614–618 (2007)

[10] Alba, E., Molina, G.: Optimal wireless sensor network layout with metaheuristics: Solving a large scale instance. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2007. LNCS, vol. 4818, pp. 527–535. Springer, Heidelberg (2008)

[11] Sengupta, S., Das, S., Nasir, M., Panigrahi, B.: Multi-objective node deployment in WSNs: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. Eng. Appl. Artif. Intell. 26(1), 405–416 (2013)

[12] Konstantinidis, A., Yang, K., Zhang, Q., Zeinalipour-Yazti, D.: A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. Comput. Networks 54(6), 960–976 (2010)

[13] Lattarulo, V., Parks, G.T.: A preliminary study of a new multi-objective optimization algorithm. In: International Conference on Evolutionary Computation (CEC), Brisbane, Australia, pp. 1–8 (2012)

[14] Lattarulo, V.: Application of an innovative optimization algorithm for the management of energy resources. BSc thesis, University of Salerno (2009)

[15] Calderaro, V., Galdi, V., Lattarulo, V., Siano, P.: A new algorithm for steady state load-shedding strategy. In: 12th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Brasov, Romania, pp. 48–53 (2010)

[16] Lattarulo, V.: Optimization of biped robot behaviors by ‘alliance algorithm’. Master’s thesis, University of Hertfordshire (2011)

[17] Lattarulo, V., van Dijk, S.G.: Application of the “Alliance algorithm” to energy constrained gait optimization. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 472–483. Springer, Heidelberg (2012)

[18] Lattarulo, V., Zhang, J., Parks, G.T.: Application of the MOAA to satellite constellation refueling optimization. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 669–684. Springer, Heidelberg (2013)

[19] Lattarulo, V., Parks, G.T.: Testing of Multi-Objective Alliance Algorithm on benchmark functions. In: GECCO 2013, Amsterdam, The Netherlands, pp. 1679–1682 (2013)

[20] Lattarulo, V., Seshadri, P., Parks, G.T.: Optimization of a supersonic airfoil using the Multi-Objective Alliance Algorithm. In: GECCO 2013, Amsterdam, The Netherlands, pp. 1333–1340 (2013)

[21] Lattarulo, V., Kipouros, T., Parks, G.T.: Application of the Multi-objective Alliance Algorithm to a benchmark aerodynamic optimization problem. In: International Conference on Evolutionary Computation (CEC), Cancun, Mexico, pp. 3182–3189 (2013)

[22] Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. 1(1), 269–271 (1959)

[23] Lattarulo, V.: Multi-Objective Alliance Algorithm. Technical Report CUED/C-EDC/TR.157, Department of Engineering, University of Cambridge (2011)

[24] Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A platform and programming language independent interface for search algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)

[25] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Trans. Evol. Comput. 7(2), 117–132 (2003)

[26] Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) (February 2006)

[27] Seshadri, A.: NSGA-II Matlab version, `http://www.mathworks.co.uk/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm` (accessed December 2012)

[28] Laumanns, M.: SPEA 2, `http://www.tik.ee.ethz.ch/pisa/selectors/spea2/?page=spea2.php` (accessed November 2011)

[29] 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)

[30] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput. 8(2), 173–195 (2000)

[31] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. TIK Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) (July 2001)

A Memetic Variant of R-NSGA-II for Reference Point Problems

Jesús Alejandro Hernández Mejía[1], Oliver Schütze[1], and Kalyanmoy Deb[2]

[1] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, C. P. 07360,
Col. San Pedro Zacatenco, Mexico City, México
jhernandez@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx
[2] College of Engineering
Michigan State University
428 S. Shaw Lane, East Lansing, MI 48824, USA
kdeb@msu.edu

Abstract. The task in multi-objective optimization is to optimize several objectives concurrently. Since the solution P of such problems is typically given by an entire set, the entire approximation of P is not always possible or even desired. Instead, it makes sense to concentrate on particular points or regions of the solution set. In case the decision maker has a certain idea about the performance of his/her product, reference point methods can be used to find the solutions that are closest to the given reference point. Evolutionary algorithms are advantageous for the treatment of such problems in particular if there are multiple reference points and/or the objectives are highly multi-modal, however, they suffer the general drawback of a slow convergence rate.

In this paper, we argue that the recently proposed Directed Search Method is suitable for an integration into evolutionary algorithms for reference point problems. We investigate the Directed Search Method for the current context and discuss its integration into the state-of-the-art algorithm R-NSGA-II. Numerical results on several benchmark problems indicate that the novel memetic strategy significantly increases the performance of its base algorithm.

Keywords: multi-objective optimization, reference point problem, memetic strategy.

1 Introduction

In many applications one is faced with the problem that several objectives have to be optimized concurrently. One important characteristic of such *multi-objective optimization problems* (MOPs) is that their solutions are typically not given by a singleton as for 'classical' scalar optimization problems (SOPs) but instead form $(k-1)$-dimensional objects, where k is the number of objectives involved in the problem.

In many cases the decision maker is interested in obtaining an approximation of the entire solution set, the so-called Pareto set, since it gives him/her the overview of all possible realizations of the project. For this, many different algorithms have been proposed so far. Among them, set based algorithms such as multi-objective evolutionary algorithms (MOEAs, see [1–3]), subdivision ([4–6]) or cell mapping techniques ([7,8])

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_17, © Springer International Publishing Switzerland 2014

have caught the interest of many researchers. Reasons for this include that these set based methods allow for the approximation of the entire set of interest in one run of the algorithm, and that they are further characterized by a great robustness and minimal requirement on the model.

In certain cases, however, it is not wanted or desired to obtain the entire solution set. Instead, selected Pareto optimal solutions are of interest to the decision maker. To obtain a single solution, e.g. scalarization methods can be used that transform the MOP into a SOP (e.g., [9–12]). If the decision maker already has a rough idea about the properties of his/her product, he/she might use *reference point methods* ([10, 13, 14]). Methods of that kind try to compute solutions that are as close as possible to a given reference point (or to a set of reference points). As for the computation of the entire Pareto set, evolutionary algorithms may be beneficial for the numerical treatment of reference point problems. A state-of-the-art algorithm that addresses this problem is the R-NSGA-II (the Reference point Non-dominated Sorting Genetic Algorithm, see [15]). R-NSGA-II can be applied successfully even if more than one reference point is given and/or if the optimization model is highly multi-modal.

In this paper, we argue that the recently proposed *Directed Search Method* (DS, see [16]) is well-suited for a hybridization with an evolutionary algorithm such as R-NSGA-II. The DS is a point-wise iterative search procedure that allows to steer the search into any direction d given in objective space. It is important to note that DS can be used with and without gradient information. The latter can be done by exploiting neighborhood information which is typically given for set based optimization strategies. Thus, the potential of DS for its hybridization with specialized evolutionary algorithms. In this study, however, we make a first attempt for the efficient integration of DS and will restrict ourselves to its gradient based version. Nevertheless, we can already observe significant improvements in the performance of the memetic strategy compared to its base MOEA on several examples.

The remainder of this paper is organized as follows: In Section 2, we briefly state the background required for the understanding of this paper. In Section 3, we present the DS for the treatment of reference point methods. In Section 4, we discuss how to integrate DS into R-NSGA-II leading to a new memetic strategy. In Section 5, we present some numerical results on some widely used benchmark models. Finally, we conclude in Section 6 and give possible paths for future work.

2 Background

A MOP can be mathematically expressed as

$$\min_{x \in Q} \{F(x)\}, \tag{MOP}$$

where $Q \subset \mathbb{R}^n$ is the domain and F is defined as the vector of the objective functions

$$F : \mathbb{R}^n \to \mathbb{R}^k, \quad F(x) = (f_1(x), \dots, f_k(x)), \tag{1}$$

and where each objective function is given by $f_i : \mathbb{R}^n \to \mathbb{R}$. Here we will handle mainly unconstrained problems (i.e., $Q = \mathbb{R}^n$), however, most of the problems considered in Section 4 are box constrained. That is, the domain is given by

$$Q = \{x \in \mathbb{R}^n \ : \ l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n\}, \tag{2}$$

where $l \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ are the lower and upper bounds, respectively.

The optimality of a MOP is defined by the concept of *dominance* ([17]): A vector $y \in \mathbb{R}^n$ is *dominated* by a vector $x \in \mathbb{R}^n$ ($x \prec y$) with respect to (MOP) if $f_i(x) \leq f_i(y)$, $i = 1, \ldots, k$, and there exists an index j such that $f_j(x) < f_j(y)$, else y is non-dominated by x. A point $x \in \mathbb{R}^n$ is called *(Pareto) optimal* or a *Pareto point* if there is no $y \in \mathbb{R}^n$ which dominates x. The set of all Pareto optimal solutions is called the *Pareto set*, and is denoted by $\mathcal{P}$. The image $F(\mathcal{P})$ of the Pareto set is called the *Pareto front*. Both sets typically form a $(k-1)$-dimensional object.

The classical reference point problem for point-wise iterative methods can be written as follows:

$$\min_{x \in Q} D(Z, F(x)), \tag{3}$$

where $Z \in \mathbb{R}^k$ is the given reference point and D is a chosen metric (we will use the Euclidean norm in this study). Since we are dealing here with entire sets of candidate solutions (populations or archives), it is advantageous to re-state the problem as:

$$\min_{\substack{A \subset Q \\ |A| = N}} dist(Z, F(A)), \tag{4}$$

where A is an archive of magnitude N, and $dist$ measures the distance of two sets as:

1. $\text{dist}(u, A) := \inf_{v \in A} D(u, v)$
2. $\text{dist}(B, A) := \sup_{u \in B} \text{dist}(u, A)$

Hereby, u and v denote points and A and B sets. The advantages of (4) over (3) in our context are that (i) archives can be considered to be 'good' even if they contain elements $a_i \in A$ that are far away from Z (those elements will not be selected by the decision maker anyway), and (ii) this concept can be extended to the case $Z = Z_1 \cup \ldots \cup Z_l$ contains multiple reference points Z_i, $i = 1, \ldots, l$.

3 Directed Search for Reference Point Problems

The DS ([16]) allows to steer the search into any direction $d \in \mathbb{R}^k$ in objective space: Given d, we are interested in a direction $\nu \in \mathbb{R}^n$ in parameter space such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + t\nu) - f_i(x_0)}{t} = d_i, \quad i = 1, \ldots, k, \tag{5}$$

where $x_0 \in \mathbb{R}^n$ is the given starting point. Denote by $J(x_0)$ the Jacobian of F at x_0,

$$J(x_0) = \left(\nabla f_1(x_0)^T, \ldots, \nabla f_k(x_0)^T\right)^T \in \mathbb{R}^{k \times n}, \tag{6}$$

then Equation (5) can in matrix vector notation be written as $J(x_0)\nu = d$. Since typically $k \ll n$, the above system can considered to be (highly) underdetermined. One suggesting choice is to use the greedy solution:

$$v_+ := J(x_0)^+ d \tag{7}$$

It is known that ν_+ solves $\min\{\|v\| \ : \ J(x_0)v = d\}$. Hence, for small values of t in the line search $x_1 := x_0 + t\nu$ the largest movement in $d-$direction is expected when choosing $v = v_+$. We note that the DS can be made derivative free via exploiting neighborhood information ([18]), however, we will not utilize this feature in this study.

In the following we discuss DS for reference point problems (3) for unconstrained MOPs. Given x_0 and Z the greedy direction d_Z in objective space is certainly given by:

$$d_Z = Z - F(x_0) \tag{8}$$

and can thus be used by the DS approach. That is, an application of DS is hence equivalent to the numerical realization of the following initial value problem (IVP):

$$\begin{aligned} &x(0) = x_0 \in \mathbb{R}^n \\ &\dot{x}(t) = J(x(t))^+(Z - F(x(t))), \quad t > 0 \end{aligned} \tag{9}$$

One appealing reason for choosing DS for the reference point problem is that we can expect (local) quadratic convergence in the special case that Z is feasible, that is, if there exists a $x^* \in Q$ such that $F(x^*) = Z$. To see this, consider the root finding problem:

$$\begin{aligned} g :&\mathbb{R}^n \to \mathbb{R}^k \\ g(x) &= F(x) - Z \end{aligned} \tag{10}$$

An iteration of the Gauss-Newton method (which converges locally quadratically, see [19]) is given by

$$\begin{aligned} x_{i+1} &= x_i - J_g(x_i)^+ g(x_i) = x_i - J(x_i)^+(F(x_i) - Z) \\ &= x_i + J(x_i)^+(Z - F(x_i)), \end{aligned} \tag{11}$$

where $J_g(x)$ denotes the Jacobian of g at x. On the other hand, an iteration performed via DS yields

$$x_{i+1} = x_i + t_i J(x_i)^+(Z - F(x_i)), \tag{12}$$

where $t_i \in \mathbb{R}_+$ is the chosen step size. Comparing (11) and (12) we see that both iterations coincide for $t_i = 1$.

It can on the other hand certainly not be expected that Z is always feasible. In order to see how to proceed in the unfeasible case, we have to understand the geometry of the solution curves of IVP (9) (see Figure 1 for the image of such a solution curve). We can divide these solution curves into two parts: In part I, a movement in d-direction is performed. Once a boundary point[1] x of (MOP) is reached, the movement is steered

[1] We can characterize a boundary point x as follows: There exists a direction d such that no movement in that direction can be performed.

along the linearized Pareto front at $F(x)$. This movement along the Pareto front constitutes part II of the solution curve. Unfortunately, the ordinary differential equation in (9) is stiff in part II which means that its numerical treatment gets complicated. To see the stiffness, let x be a boundary point. Then there exists a direction d such that the equation $J(x)\nu = d$ has no solution. This is equivalent to $rank(J(x)) < k$ which means that the condition number of $J(x)$ is infinite.

In order to overcome this stiffness, we can perform a linearization to steer the search directly as follows: Let x be a boundary point with weight $\alpha \in \mathbb{R}^k$ such that

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x) = 0. \tag{13}$$

Further, let a QR factorization of α be given,

$$\alpha = QR = (q_1, \ldots, q_k)R. \tag{14}$$

Then the column vectors of $Q_2 = (q_2, \ldots, q_k) \in \mathbb{R}^{k\times(k-1)}$ form an orthonormal basis of the linearized Pareto front at $F(x)$. Given the direction d_Z, the projection onto the tangent space is thus given by

$$d_{new} = Q_2(\underbrace{Q_2^T Q_2}_{=I})^{-1} Q_2^T d_Z = Q_2 Q_2^T d_Z. \tag{15}$$

Denote by $P(x) := Q_2(x)Q_2(x)^T$ the projection described above. Then we suggest to solve the following IVP for points x_0 where $F(x_0)$ is on the boundary of the image

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= J(x(t))^+ P(x(t))(Z - F(x(t))), \quad t > 0 \end{aligned} \tag{16}$$

The switch between (9) and (16) can be handled via monitoring the condition number of $J(x)$: If $J(x) < tol$ for a certain tolerance value we suggest to choose to follow the flow in (9), else (16) should be taken.

It remains to select the step sizes in order to obtain efficient numerical realizations of (9) and (16) which is subject of ongoing research. For the implementations used for this study, we have followed the step size control of the DS Descent Method described in [20] together with the initial step size $t_0 = 1$ motivated by the above discussion. For the numerical realization of (16) we have adapted the step size control of the DS Continuation Method which is also described in [16].

4 Integrating DS into R-NSGA-II

In the following we describe a possible integration of DS into R-NSGA-II, however, we would like to stress that a similar integration can in principle be done into any other set based reference point method. We will briefly describe R-NSGA-II as well as where we decided to apply local search (LS) and the reason for it.

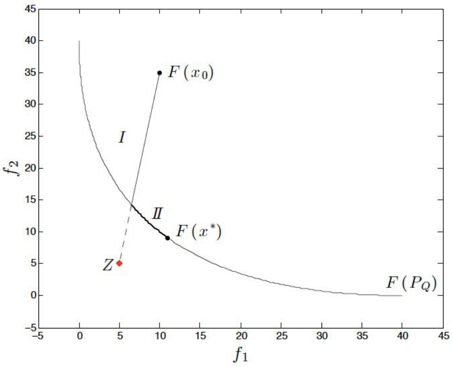


Fig. 1. Typical image of a solution curve of (9)

In one iteration of R-NSGA-II parent and offspring population are combined and non dominated sorting is performed to obtain different levels of non dominated fronts. Analogously to NSGA-II, the first non dominated fronts will be chosen to be included into the next parent population. However, since the last front could not be included but partially, a modified crowding distance is associated to each solution in order to decide which individuals to take. The new crowding distance is the lowest hierarchy the solution has in the sorted lists of distances to each reference point, being hiearchy one the closest solution, two the second closest and so on. A parameter ϵ is introduced to control the spread of solutions in the final archive. We decided to apply LS on the parent population before an iteration starts, hence, improved solutions are emphasized and are expected to create new solutions similar to them in the next generations.

As for the LS algorithm integration within a memetic strategy, many aspects have to be considered. Among the main (generic) parameters are: The number of individuals to which LS will be applied, the maximal iteration number (depth) of the LS applied to an individual, and frequency of the same.

Next to these generic parameters—see Table 1 for the choices we made for our implementations—there are some more specific ones. Among them, the decision of which solutions will be chosen for improvement is an important aspect. Considering R-NSGA-II, one could try to improve individuals on the first non dominated front and backwards from then on. This approach, however, is not useful in particular in higher dimensional spaces since most of the individuals will belong to this front. A second (and more promising) alternative is to improve solutions which are already close to reference points whether they are dominated or not. This arises the new problem of which reference points to be taken for improvement: Considering two or more reference points, it is likely that the closest solutions within the population are not the same (and even if they were), hence, some reference points would gain more attention than others. We propose a possible solution as follows: Assign an integer identifier to each reference point and store the index of the last reference point improved. Starting from improving the first reference point, switch to improve the next one (by modular addition) until the number of individuals designated for local search is reached.

Another point is that DS—as well as all other local search engines—can get stuck on regions which are not globally optimal. In order to prevent wasting function evaluations on currently non-improvable reference points, one last parameter will be used: The maximum number of improvement trails toward a given reference point.

As local search we use the numerical realization of DS explained in the previous section with one exception: Following the suggestion made in [21] we will perform a greedy search toward the Pareto set in case the individual x_0 is 'far away' from the solution set which can be monitored by the angle between the objectives gradients.

Finally, we note that the feasibility of reference points represent a potential problem for the DS approach. Though it makes sense to present to the decision maker better (i.e., dominating) solutions than for a given feasible reference point—as e.g. done in R-NSGA-II—it is ad hoc unclear in which direction to perform the movement. Further, it does not comply with the problem statement in (4) since optimal solutions for (4) are ones those image coincides with the reference point. Here, we will simply stop the movement toward a reference point once feasibility has been detected and current solutions are near enough.

Having stated this we are in the position to explain the hybrid RDS-R-NSGA-II (compare to Algorithm 1):

1. Compute the distance of each member of the population to each reference point in ascending order. (Line 1)
2. Mark the feasible reference points found so far as non improvable. Set a counter of solutions improved so far to zero. (Lines 2-3)
3. Cycle through the set of reference points (starting from the last one used, or one if this is the first call) until an improvable reference point is found. Set a test counter variable to zero and a direction vector as the empty set. (Lines 4-8)
4. If the test counter reaches the maximum tests, mark the current reference point as non improvable and continue to Step 3, else increment the test counter. (Line 10)
5. Obtain the next closest solution for the current reference point and set it as the working iterate, compute the Jacobian and the best descent direction. (Lines 11-21)
6. Set the initial step size as one. Modify the direction and initial step properly in order to avoid unfeasible regions. (Lines 23-24)
7. If the norm of the direction or step size are too small, discard the current iterate and continue in step 5, else, perform a search in the given direction. (Lines 25-26)
8. If the new solution is not better than the original and the Jacobian is available, compute an optimal step by approximating a second order polynomial. (Line 27)
9. Exchange the new solution for the one in the population if it is closer to the current reference point and continue the search in the given direction until the maximum iterations are reached or there is no further gain, else, continue in step 5. (Lines 28-30)
10. Stop the process if the number of sought improved solutions is reached or if the set of improvable reference points is empty. Go to Step 3 in any other case. (Line 33)

5 Numerical Results

In the following we will present some numerical results where we concentrate on models with small number of objectives (up to five). Table 1 shows the parameter values we

Algorithm 1. Algorithm RDS-R-NSGA-II

```
Require: Population $P$, Ref. pts $Z_1, ..., Z_l$, Feasible ref. pts. so far $i \in \{1, ..., l\}$, local search max iterations $lsd$, solutions to improve $nsi$, last reference point used $lz$, maximum tests on a ref. pt. $maxTst$
Ensure: Population with at most $nsi$ improved solutions $P'$, index of last reference point $lz$.
For each $Z_i$ sort $P$ using a chosen norm.
Set $solsImproved := 0$ ; $done :=$ false ; $improvableRefPts := \{1, ..., l\}$ .
Remove feasible reference points indices from $improvableRefPts$.
while not $done$ do
  if $lz \notin improvableRefPts$ and $improvableRefPts \neq \emptyset$ then
    $lz := (lz\%l) + 1$, Continue;
  end if
  Set $\nu := \{\}$; $tstCntr := 0$
  for $j := 1$ to $|P|$ do
    if $tstCntr > maxTst$ then remove $lz$ from $improvableRefPts$, break for loop else set $tstCntr := tstCntr + 1$ end if
    Set $x_0$ and $F_0$ as the decision and objective vector respectively of the $j$-th closest solution to $Z_{lz}$ and $lzIters := 0$.
    if $\nu = \{\}$ then
      Compute the Jacobian $J \in \mathbb{R}^{k \times n}$
      if $k = 2$ and $\angle(\nabla f_1, \nabla f_2) < 20\circ$ then
        Compute $\nu = -\frac{1}{2}(\frac{\nabla f_1}{||\nabla f_1||} + \frac{\nabla f_2}{||\nabla f_2||})$. {Lara descent direction}
      else
        Set $d = Z_i - F_i$.
        Compute $\nu = J^{+}d$ and set $\nu = \frac{\nu}{||\nu||}$ {DS descent direction}
      end if
      Set $gradAvailable :=$ true.
    end if
    while true do
      Set $t_0 := 1$.
      Call $BoxHandling$ to get $\nu_h$ and $t_h$.
      if $t_h$ or $||\nu_h||$ are too small then break while loop end if
      Set $x_1 := x_0 + t_h nu_h$ and $F_1 := F(x_1)$.
      if $||F_1 - Z_{lz}|| > ||F_0 - Z_{lz}||$ and $gradAvailable$ then compute $t_{opt}$ using $F_0$ , $F_1$ and $J$. Update $x_1$, $F_1$. end if
      if $||F_1 - Z_{lz}|| > ||F_0 - Z_{lz}||$ or $lzIters = lsd$ then break while loop. end if
      Exchange $j$-th individual's decision and objective vector with $x_1$ and $F_1$ respectively.
      Set $x_0 := x_1$, $F_0 := F_1$, $lzIters := lzIters + 1$ and $gradAvailable :=$ false.
    end while
  end for
  if $solsImproved = nsi$ or $improvableRefPts = \emptyset$ then set $done :=$ true else $lz := (lz\%l) + 1$.end if
end while
```

have chosen for each algorithm, R-NSGA-II and its hybrid RDS-R-NSGA-II. For a fair comparison, both algorithms have the same function budget. In addition, LS function calls are also subtracted from the function budget. In all cases we have only taken unfeasible reference points. In order to assess the performances, we have chosen to take the IGD indicator ([22]) applied to reference problems (denote by IGD_Z) which can be viewed as an averaged version of the distance measurement used in (4):

$$IGD_Z(F(A), Z) = \frac{1}{|Z|}\sum_{i=1}^{|Z|} \min_{j=1}^{|A|} dist(Z_i, F(a_j)) \qquad (IGD_Z)$$

Hereby, $A \subset Q$ is the given population or archive and $dist$ the chosen distance metric (for this, we have taken the Euclidean norm).

Table 1. Chosen parameter values for the different algorithms. Hereby, we use the following notation: N_{pop} denotes the population size, Gen the number of generations, P_X and P_M the probability for crossover and mutation, LS freq and LS dep the frequency and maximum iterations for local search, StI the number of solutions to improve and Max LSC the maximum number of LS Calls.

	R-NSGA-II							RDS			
	N_{pop}	Gen	P_X	P_M	ϵ	η_c	η_m	LS freq	LS dep	StI	Max LSC
F_1	100	100	0.9	0.033	5E-6	15	20	5	30	1	9
$ZDT1$	100	100	0.9	0.033	5E-6	15	20	10	30	5	4
$ZDT2$	100	100	0.9	0.033	5E-6	15	20	10	30	6	4
$ZDT3$	100	100	0.9	0.033	5E-4	15	20	5	30	2	10
$DTLZ2, k = 3$	100	100	0.9	0.083	1E-3	15	20	5	30	1	5
$DTLZ2, k = 5$	100	100	0.9	0.083	1E-3	15	20	5	30	2	5
$DTLZ3, k = 3$	100	150	0.9	0.1	1E-3	15	20	15	30	3	8

First we consider the bi-objective problem CONV ([23])

$$f_1(x) = ||x - a_1||_2^2, \quad f_2(x) = ||x - a_2||_2^2, \qquad (17)$$

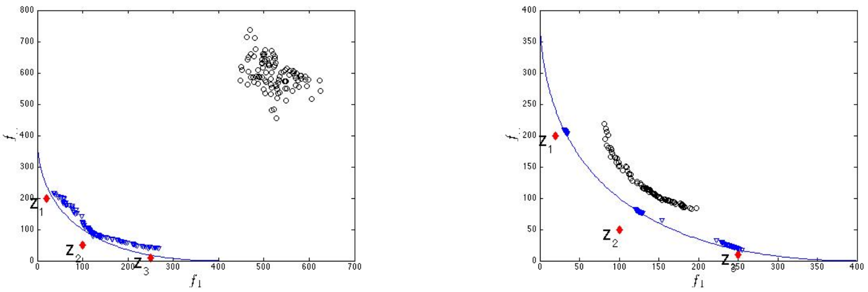
where $a_1 = (1, ..., 1)^T \in \mathbb{R}^{100}$ and $a_2 = -a_1$, and where $Q = \mathbb{R}^{100}$ is the domain. We chose the three reference points $Z = \{(20, 200)^T, (100, 50)^T (250, 10)^T\}$. Since CONV is a convex problem large improvements are expected via the help of local search. This is indeed the case as can be seen in Figure 2 as well as in Table 2, where the IGD_Z values for all the examples considered in this study are presented obtained after 20, 40 and 100 percent of the given function budget.

Next, we consider MOP ZDT1 from the ZDT benchmark suite ([24]) using $n = 30$ (as well as for all remaining MOPs) and $Z = \{(0.1, 0.6)^T, (0.5, 0.2)^T, (0.9, 0)^T\}$. Compared to CONV, RDS-R-NSGA-II needs more function evaluations on ZDT1 to obtain a covering of the optimal solutions near the reference points, but comes quite close after already 40 percent of the given function budget (compare to Figure 3 and Table 2).

ZDT2 represents a challenge for evolutionary algorithms since typically in first stages of the search all solutions concentrate on the left top area of the objective space. Here, we can see in Figure 4 that RDS-R-NSGA-II achieves a focus on the desired areas here given by $Z = \{(0.1, 0.9)^T, (0.8, 0.2)^T, (0.6, 0.5)^T\}$ after 40 percent of the function budget. At the end of the given function budget, R-NSGA-II still performs a movement along the Pareto front while its memetic variant is covering the optimal areas.

ZDT3 has a more complex geometry than the previous ones, where not all boundary solutions are optimal. Here we have chosen $Z = \{(0.8, -0.6)^T, (0.1, 0.6)^T, (0.35, 0.1)^T\}$. At twenty percent of function budget, the memetic strategy is outperforming R-NSGA-II (compare to Figure 5). However, it has to be noted that some optimal solutions near Z_1 could hardly be reached since a directed search may lead outside the feasible region. Nevertheless, at hundred percent solutions of both algorithms are very close together.

Analogously, we have tested the two methods on two three-objective problems—DTLZ2 and DTLZ3 taken from [25], see Figures 6 and 7—as well as on DTLZ2 with $k = 5$ objectives yielding a similar evolution of the performance.

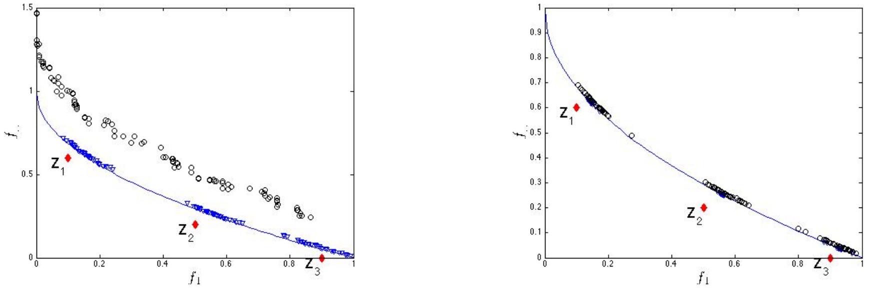


(a) 20% of budget

(b) 100% of budget

Fig. 2. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on CONV

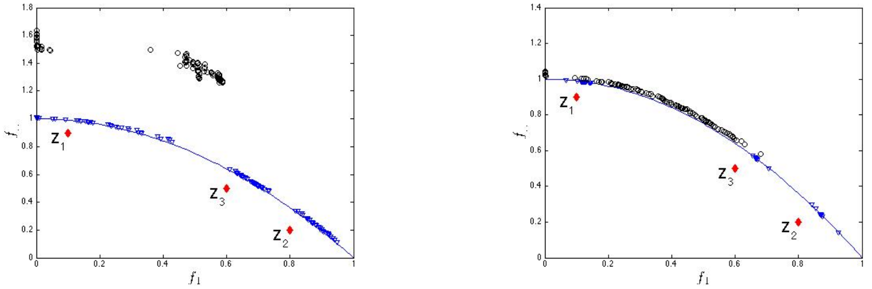
Considering these seven test examples we see from Table 2 that RDS-R-NSGA-II wins over its base algorithm in seven out of seven cases when we stop the search after 20 percent of the function budget and still in all 7 cases after 40 percent. The more the search continues, the more both methods converge toward the optimal regions. Hence, both methods get solutions near to the optimum after the given function budget. However, it is apparent that the memetic strategy yields faster convergence. In fact, using RDS-R-NSGA-II, solutions that are nearly optimal can already be reached after 40 percent of the given budget in most cases. Thus, one can say that RDS-R-NSGA-II accomplishes its task.



(a) 40% of budget

(b) 100% of budget

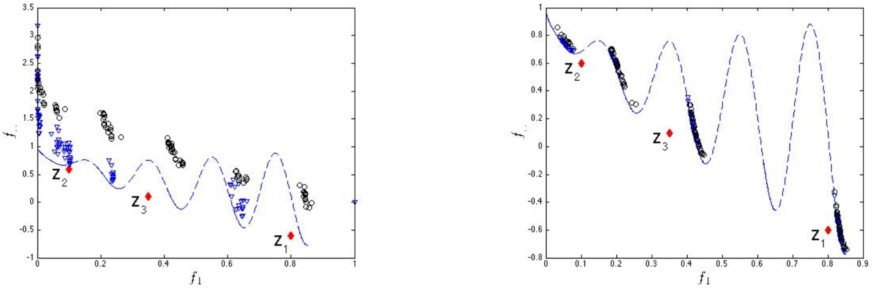
Fig. 3. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on ZDT1



(a) 40% of budget

(b) 100% of budget

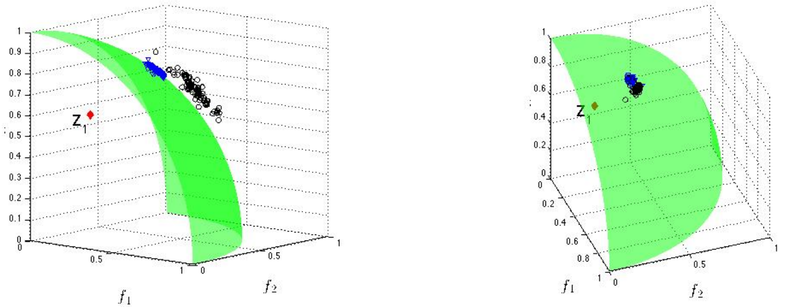
Fig. 4. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on ZDT2



(a) 20% of budget

(b) 100% of budget

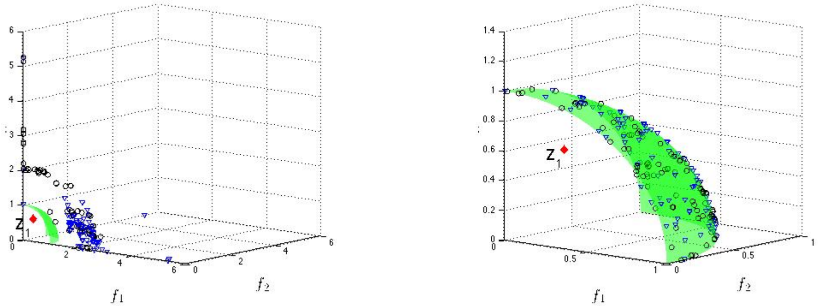
Fig. 5. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on ZDT3



(a) 40% of budget

(b) 100% of budget

Fig. 6. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on DTLZ2



(a) 20% of budget

(b) 100% of budget

Fig. 7. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on DTLZ3

Table 2. IGD_Z values for the two different algorithms on the considered reference point problems. The results are averaged over 30 independent runs.

	RDS-R-NSGA-II			R-NSGA-II		
	mean (std. dev)					
	20	40	100	20	40	100
F1	31.107 (2.019)	21.633 (1.386)	20.190 (0.627)	559.465 (43.855)	221.012 (19.335)	74.720 (4.789)
ZDT1	0.177 (0.118)	0.062 (0.010)	0.056 (0.000)	0.565 (0.063)	0.224 (0.041)	0.063 (0.002)
ZDT2	0.164 (0.063)	0.091 (0.013)	0.085 (1E-4)	1.346 (0.212)	0.734 (0.264)	0.1764 (0.112)
ZDT3	0.247 (0.124)	0.099 (0.037)	0.068 (0.019)	0.416 (0.051)	0.137 (0.049)	0.069 (0.011)
DTLZ2 $k = 3$	0.359 (0.021)	0.358 (0.014)	0.344 (0.005)	0.583 (0.043)	0.427 (0.030)	0.348 (0.004)
DTLZ2 $k = 5$	0.475 (0.025)	0.463 (0.012)	0.473 (0.011)	0.675 (0.048)	0.503 (0.021)	0.474 (0.011)
DTLZ3	0.530 (0.310)	0.374 (0.029)	0.346 (0.009)	0.910 (0.598)	0.417 (0.186)	0.346 (0.007)

6 Conclusions and Future Work

In this paper, we have addressed the numerical treatment of reference point problems by means of a memetic strategy. For this, we have discussed and adapted the recently proposed Directed Search (DS) method that allows to steer the search into any direction given in objective space. Further, we have made a first attempt to integrate the DS into R-NSGA-II, a state-of-the-art evolutionary algorithm designed for reference point problems. First results where we have focussed on multi-objective problems with a low number of objectives ($2 \leq k \leq 5$) indicate that such a hybrid leads to a significant speed up of the computations in almost all cases.

Though the first results are very promising, there are, however, many points to be addressed in the future. This includes the more thorough discussion of the DS for the problem at hand as well as an improvement of its numerical realization. Next, the interplay of local and global search can be further enhanced for a better performance in particular for multi-modal problems. Finally, we will consider problems with more objectives since in that case the advantage over classical methods—i.e., the approximation of the entire solution set—will become more significant.

Acknowledgement. The first author acknowledges support from CONACyT through a scholarship to pursue undergraduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 128554.

References

[1] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X

[2] Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research (2006)

[3] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007) ISBN 978-0-387-33254-3

[4] Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. Journal of Optimization Theory and Applications 124, 113–155 (2005)

[5] Jahn, J.: Multiobjective search algorithm with subdivision technique. Computational Optimization and Applications 35(2), 161–175 (2006)

[6] Schütze, O., Vasile, M., Junge, O., Dellnitz, M., Izzo, D.: Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. Engineering Optimization 41(2), 155–181 (2009)

[7] Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., Sun, J.Q.: Simple cell mapping method for multi-objective optimal feedback control design. International Journal of Dynamics and Control 1(3), 231–238 (2013)

[8] Hernández, C., Sun, J.-Q., Schütze, O.: Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 171–188. Springer, Heidelberg (2013)

[9] Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization 8, 631–657 (1998)

[10] Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)

[11] Fliege, J.: Gap-free computation of Pareto-points by quadratic scalarizations. Mathematical Methods of Operations Research 59, 69–89 (2004)

[12] Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Heidelberg (2008) ISBN 978-3-540-79157-7

[13] Ignizio, J.: Goal programming and extensions. Lexington books. Lexington Books (1976)

[14] Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Application. Lecture Notes in Economics and Mathematical Systems, vol. 177, pp. 468–486. Springer, Heidelberg (1980)

[15] Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research, 635–642 (2006)

[16] Schütze, O., Lara, A., Coello, C.A.C.: 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)

[17] Pareto, V.: Manual of Political Economy. The MacMillan Press (1971 (original edition in French in 1927))

[18] Lara, A., Alvarado, S., Salomon, S., Avigad, G., Coello, C.A.C., Schütze, O.: The gradient free directed search method as local search within multi-objective evolutionary algorithms. In: Schütze, O., Coello Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., Legrand, P. (eds.) EVOLVE - A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation II. AISC, vol. 175, pp. 153–168. Springer, Heidelberg (2012)

[19] Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer (1990)

[20] 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 Automatic Control (CCE 2010), pp. 395–399 (2010)

[21] Hernández, V.A.S., Schütze, O., Rudolph, G., Trautmann, H.: The directed search method for pareto front approximations with maximum dominated hypervolume. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 189–205. Springer, Heidelberg (2013)

[22] 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)

[23] Köppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)

[24] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)

[25] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, pp. 105–145. Springer, USA (2005)

A Multiobjective Evolutionary Algorithm Guided by Averaged Hausdorff Distance to Aspiration Sets

Günter Rudolph[1], Oliver Schütze[2], Christian Grimme[3], and Heike Trautmann[3]

[1] Department of Computer Science, TU Dortmund University, Germany
guenter.rudolph@tu-dortmund.de
[2] Department of Computer Science, CINVESTAV, Mexico City, Mexico
schuetze@cs.cinvestav.mx
[3] Department of Information Systems, University of Münster, Germany
{christian.grimme,trautmann}@uni-muenster.de

Abstract. The incorporation of expert knowledge into multiobjective optimization is an important issue which in this paper is reflected in terms of an aspiration set consisting of multiple reference points. The behaviour of the recently introduced evolutionary multiobjective algorithm AS-EMOA is analysed in detail and comparatively studied for bi-objective optimization problems w.r.t. R-NSGA2 and a respective variant. It will be shown that the averaged Hausdorff distance, integrated into AS-EMOA, is an efficient means to accurately approximate the desired aspiration set.

Keywords: multi-objective optimization, aspiration set, preferences.

1 Introduction

In multiobjective evolutionary optimization the *a posteriori* approach is predominant: At first the evolutionary multiobjective algorithm (EMOA) approximates the Pareto front with a finite number of candidate solutions before the decision maker (e.g. product designer) selects the solution with the most desired properties to be realized. If these desires are known in advance (although these desires may be utopian because they cannot be achieved technically) one may apply a so-called reference method to find a solution that matches these desires as close as possible. In this case one can save much computational effort because it is no longer necessary to approximate the entire Pareto front but only a small part of it. This remains true if the decision maker is a little bit uncertain about his/her desires so that a finite set of reference points is specified instead of a single one.

Here, we propose an evolutionary method to approximate only desired parts of the PF (which we call *aspiration set*) that is a marriage between a set-based version of the original reference point method [1] and the *averaged* Hausdorff distance [2] as selection criterion[1].

[1] A preliminary version of this work appeared in [3]. The current work, however, extends those preliminary results by a comprehensive experimental evaluation and a comparison with state-of-the art reference point approaches.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_18, © Springer International Publishing Switzerland 2014

Although many other distance measures are possible we rely on the averaged Hausdorff distance because in our previous work [4–7] we successfully used the concept of the averaged Hausdorff distance in designing EMOAs that find an evenly spaced approximation of the PF.

The remainder is organized as follows: section 2 introduces basic facts and notations about multiobjective optimization, reference point methods, aspiration sets, and averaged Hausdorff distance. The new evolutionary algorithm and two competitors are described in section 4, before we detail experimental setup and results in section 5. Finally, we summarize our findings and provide an outlook regarding further research in section 6.

2 Foundations

2.1 Multiobjective Optimization

In the following we consider unconstrained multiobjective optimization problems (MOPs) of the form $\min\{f(x) : x \in \mathbb{R}^n\}$ where $f(x) = (f_1(x), \ldots, f_d(x))'$ is a vector-valued mapping with $d \geq 2$ objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, d$ that are to be minimized simultaneously. The optimality of a MOP is defined by the concept of *dominance*.

Let $u, v \in F \subseteq \mathbb{R}^d$ where F is equipped with the partial order $\preceq$ defined by $u \preceq v \Leftrightarrow \forall i = 1, \ldots d : u_i \leq v_i$. If $u \prec v \Leftrightarrow u \preceq v \wedge u \neq v$ then v is said to be *dominated by* u. An element u is termed *nondominated* relative to $V \subseteq F$ if there is no $v \in V$ that dominates u. The set $\mathsf{ND}(V, \preceq) = \{u \in V \mid \nexists v \in V : v \prec u\}$ is called the *nondominated set* relative to V.

If $F = f(X)$ is the objective space of some MOP with decision space $X \subseteq \mathbb{R}^n$ and objective function $f(\cdot)$ then the set $F^* = \mathsf{ND}(f(X), \preceq)$ is called the *Pareto front* (PF). Elements $x \in X$ with $f(x) \in F^*$ are termed *Pareto-optimal* and the set X^* of all Pareto-optimal points is called the *Pareto set* (PS). Moreover, for some $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}^d$ the set $\mathsf{ND}_f(X, \preceq) = \{x \in X : f(x) \in \mathsf{ND}(f(X), \preceq)\}$ contains those elements from X whose images are nondominated in image space $f(X) = \{f(x) : x \in X\} \subseteq \mathbb{R}^d$.

2.2 Reference Point Problems

In some cases one is not interested in finding an approximation of the entire PF but rather in particular (optimal) solutions. For instance, if the decision maker has a certain idea about the target features of his/her product, one can use *reference point methods* [1, 8] to find a solution that is closest to a so-called reference point gathering the user-given level of aspiration for each objective. A general reference point problem is stated as follows:

$$\min_{x \in X} d(z, f(x)), \tag{1}$$

where $z \in \mathbb{R}^d$ is the given reference point and $d(\cdot,\cdot)$ is a chosen metric. Note that even if z is not feasible (i.e., there exists no $x \in X$ such that $f(x) = z$) it does not necessarily hold that a solution of (1) is Pareto optimal. If this is desired, one can modify (1) to

$$\min_{x \in X^*} d(z, f(x)), \tag{2}$$

Approaches guided by single reference points and aiming for a single desired solution on the Pareto front can be found in [9] and more are summarized in [10]. An interesting variant is R-NSGA2 [11]. This approach does not only provide a single solution but also some additional solutions in its neighborhood, whereas multiple reference points can be used to approximate larger parts of the PF by running the original method in parallel for each reference point [12]. In section 4 these approaches are discussed for experimental comparison reasons in section 5.

2.3 Averaged Hausdorff Measure

In this study we need to measure the distance between two sets. One natural choice would be the Hausdorff distance (e.g., [13]) which is defined as follows:

Definition 1. *The value* $d_H(A,B) := \max(d(A,B), d(B,A))$ *is termed the* Hausdorff distance *between two sets* $A, B \subset \mathbb{R}^n$, *where*

$$d(B,A) := \sup\{d(u,A) : u \in B\} \;\; \textit{and} \;\; d(u,A) := \inf\{\|u - v\| : v \in A\}$$

for $u, v \in \mathbb{R}^n$ *and a vector norm* $\|\cdot\|$.

The Hausdorff distance is widely used in many fields. It has, however, certain limitations for our purpose: One of the sets considered in $d_H(A,B)$ will be the outcome of an EMOA and may contain outliers. These would be punished too strongly by d_H. As a remedy, we follow the suggestion of [2] and use the *averaged* Hausdorff distance.

Definition 2. *The value* $\Delta_p(A,B) = \max(\mathsf{GD}_p(A,B), \mathsf{IGD}_p(A,B))$ *with*

$$\mathsf{GD}_p(A,B) = \left(\frac{1}{|A|}\sum_{a \in A} d(a,B)^p\right)^{1/p} \;\; \textit{and} \;\; \mathsf{IGD}_p(A,B) = \left(\frac{1}{|B|}\sum_{b \in B} d(b,A)^p\right)^{1/p}$$

for $p > 0$ *is termed the* averaged Hausdorff distance *between sets* A *and* B.

The indicator Δ_p combines slight variations of the two well-known distance measures Generational Distance (GD) and the Inverted Generational Distance (IGD), see [14]. It is $\Delta_\infty = d_H$, but for finite values of p the indicator Δ_p averages the distances considered in d_H. Thus, in contrast to d_H, Δ_p in particular does not punish single (or few) outliers in a candidate set. We consider discrete (or discretized) sets, for a formulation of Δ_p, for general sets we refer to [2].

3 Problem Statement and Choice of the Distance Function

In the following we extend the idea of reference *point* problems to the context of reference *sets* or so-called *aspiration sets* and state the resulting optimization problem.

Assume we are given a reference set $R \subset \mathbb{R}^d$ (consider e.g., two different reference points or the line segment that connects them, but more general sets can be taken as well). Then the task is to find a finite archive $A \subset X^*$ that contains N elements whose image $f(A)$ is as 'close' as possible to R. Given a distance function $d(\cdot,\cdot)$ the problem can hence be formulated mathematically as

$$\min_{\substack{A \subset X^* \\ |A|=N}} d(f(A), R) \tag{3}$$

It remains to determine the distance function $d(\cdot,\cdot)$ in (3). For this, we will use Δ_p (here, $p = 1$) together with the following weighting that is done in objective space: For a vector x and corresponding function values $f(x)$ for a bi-objective problem we define the map $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^2$ as follows:

$$\tilde{f}_1(x) = \frac{f_1(x) - min_1}{max_2 - min_2}, \quad \tilde{f}_2(x) = \frac{f_2(x) - min_2}{max_1 - min_1}, \tag{4}$$

where

$$min_j = \min\{y_j \ : \ y \in R\}, \quad max_j = \max\{y_j \ : \ y \in R\}, \quad j = 1, 2. \tag{5}$$

Thus, given the reference set R and the number N of elements in the population (archive) the optimization problem we focus in this work is

$$\min_{\substack{A \subset X^* \\ |A|=N}} \Delta_p(\tilde{f}(A), R) \tag{6}$$

The motivation behind this scaling is that we aim at an outcome set A such that $f(A)$ is distributed roughly 'in parallel' to R. This is particularly interesting if the values of the objective functions (and hence $max_j - min_j$) are in different ranges. Figure 1 shows an example where we see the optimal 10-element archive for both f and $\tilde{f}$ used in (6). We note, however, that certainly other choices are possible depending on the particular application. Further, note that the weighting is (so far) only applicable to bi-objective problems. Both points will be addressed in future work.

4 Algorithms

In this study we compare the recently developed evolutionary algorithm guided by aspiration sets, called AS-EMOA, with the state-of-the-art competitor from the literature, the so-called R-NSGA2, and its multistart version, denoted R-NSGA2(M).

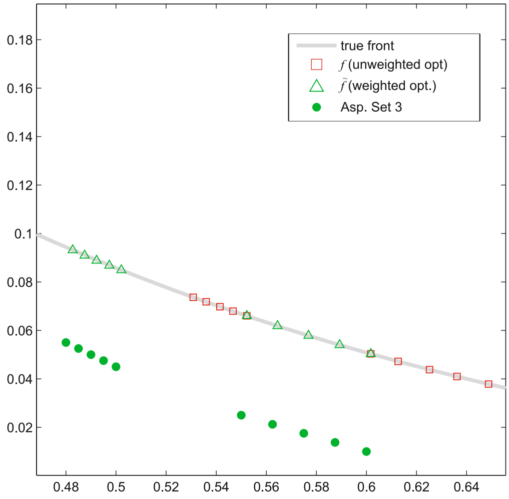


Fig. 1. Aspiration set and optimal 10-element archives of the scaled and the unscaled problem

4.1 AS-EMOA

The AS-EMOA was designed for approximating the aspiration set: We apply the weighted normalization for each candidate solution as described in the previous section in order to focus on the given aspiration set and to avoid biases due to its orientation in objective space. Thus, in line 4 of the Δ_1-update algorithm given below we first apply the normalization before determining the values $h(a)$ for $a \in A$.

AS-EMOA

```
Require: aspiration set $R$
initialize population $P$ with $|P| = \mu$
$P = \mathsf{ND}_f(P, \preceq)$
while termination criterion not fulfilled do
    generate offspring $x$ by variation of parents from $P$
    $P = \Delta_1$-update$(P, x; R)$
end while
```

Δ_1-update (line 8: ties are broken at random)

```
Require: archive set $A$, new $x$, aspiration set $R$
 1: $A = \mathsf{ND}_f(A \cup \{x\}, \preceq)$
 2: if $|A| > N_R := |R|$ then
 3:     for all $a \in A$ do
 4:         $h(a) = \Delta_1(\tilde{f}(A \setminus \{a\}), R)$
 5:     end for
 6:     $A^* = \{a^* \in A : a^* = \mathsf{argmin}\{h(a) : a \in A\}\}$
 7:     if $|A^*| > 1$ then
 8:         $a^* = \mathsf{argmin}\{\mathsf{GD}_1(A \setminus \{a\}, R) : a \in A^*\}$
 9:     end if
10:     $A = A \setminus \{a^*\}$
11: end if
```

4.2 R-NSGA2 Variants for Comparison

R-NSGA2 proposed by Deb et al. [11] is one of few preference-based EMOAs which work without weighting or modifications of dominance relations. Instead it strives for a user-provided reference point. Furthermore, it can be extended to a reference set approach using aspiration sets instead of single reference points. Here, we use the original

R-NSGA2 and a modified variant to deal with aspiration sets for comparison reasons with AS-EMOA. In the following we give a brief description of R-NSGA2 and our modification.

R-NSGA2: The approach is based on NSGA2 [15] and applies non-dominated sorting on combined parent and offspring populations. Solutions for the next generation are then selected front-wise from the best fronts determined. The given reference point information is used inside a modified crowding distance operator when only some solutions from a front level can be moved to the next generation in order to keep population size constant. In this modified procedure crowding distance results from the euclidean distance of each solution to the reference point. For multiple reference points the minimum crowding distance over all reference points is preserved. In the following selection process, solutions with smaller crowding distance are preferred. Further, to control diversification of selected solutions, a parameter ϵ is introduced to group all solutions having a normalized difference less than ϵ between them over all objective values. From such a group a single random representative is preserved. The remaining solutions are assigned a large crowding distance which makes them unattractive for further consideration.

R-NSGA2(M): As the before described standard approach of R-NSGA2 is originally focused on a single or few far apart reference points and merely considers the minimum distance of a solution to one reference point out of a relatively dense but structured aspiration set, we modified it to respect all points from such a set equally. Therefore, we start multiple instances of the standard R-NSGA2 in parallel with equal fractions of the total function evaluations each. Finally, we aggregate the resulting solution sets and apply domination filtering for receiving the final solution set.

5 Experiments

In order to proof the concept of the proposed AS-EMOA and evaluate its performance, we perform simulation runs with four bi-objective standard test problems from the MOO domain. For comparison reasons we use the aforementioned state-of-the-art approach R-NSGA2 and the slightly modified, concurrent variant R-NSGA2(M).

5.1 Experimental Setup

AS-EMOA, R-NSGA2, and R-NSGA2(M) were evaluated for four well known bi-objective test problems (SPHERE [4]: convex, $n = 2$, DTLZ2: concave, $n = 10$, DENT [16]: convex-concave, $n = 2$, ZDT3: disconnected, $n = 20$) which expose different characteristics regarding convexity/concavity, locality and search space dimensionality. For a full definition refer also to [4]. The applied aspiration sets were generated in the utopian objective space (”before PF”) and in the dominated objective space (”behind PF”), see Figures 2 and 3. All algorithms were executed 20 times per test problem and considered aspiration sets for population sizes of $\mu \in \{10, 20\}$ and 50,000 function evaluations (FE).

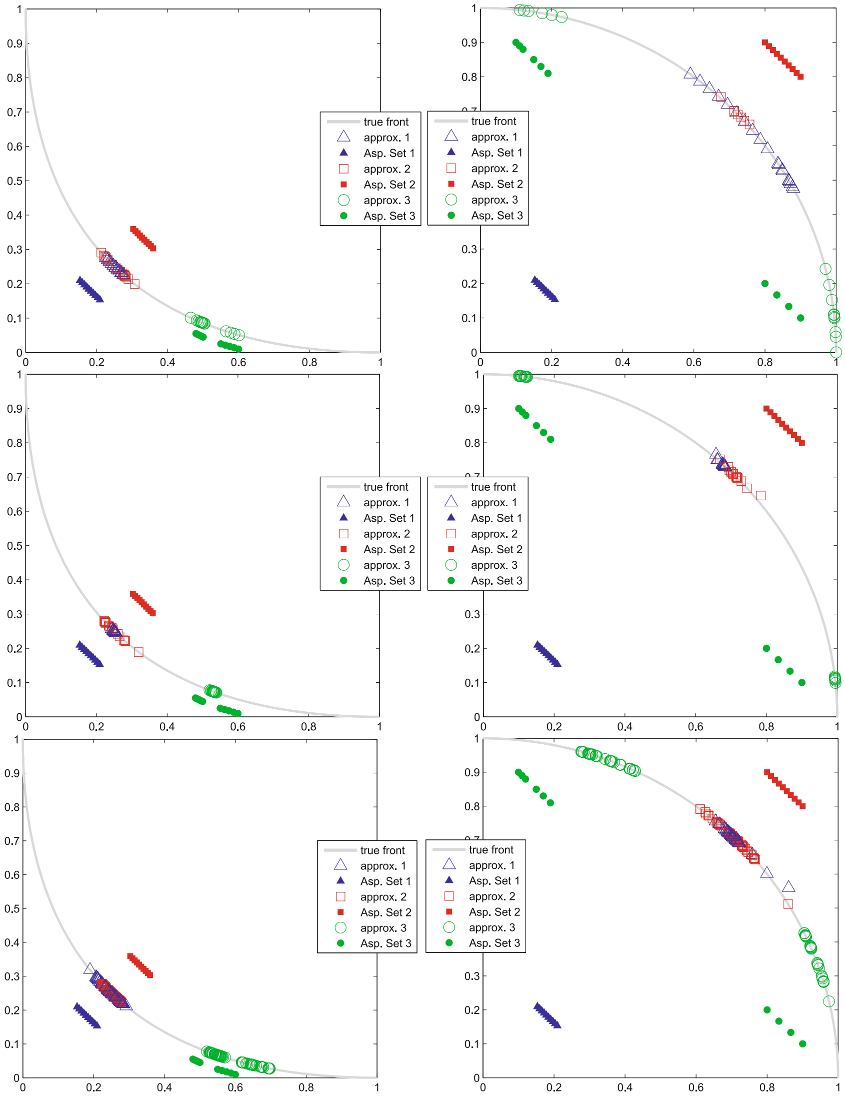


Fig. 2. Results of AS-EMOA (top), R-NSGA2 (middle) and R-NSGA2(M) (bottom) for SPHERE (left) and DTLZ2 (right) using various reference sets and $p = 1$, $\mu = 20$

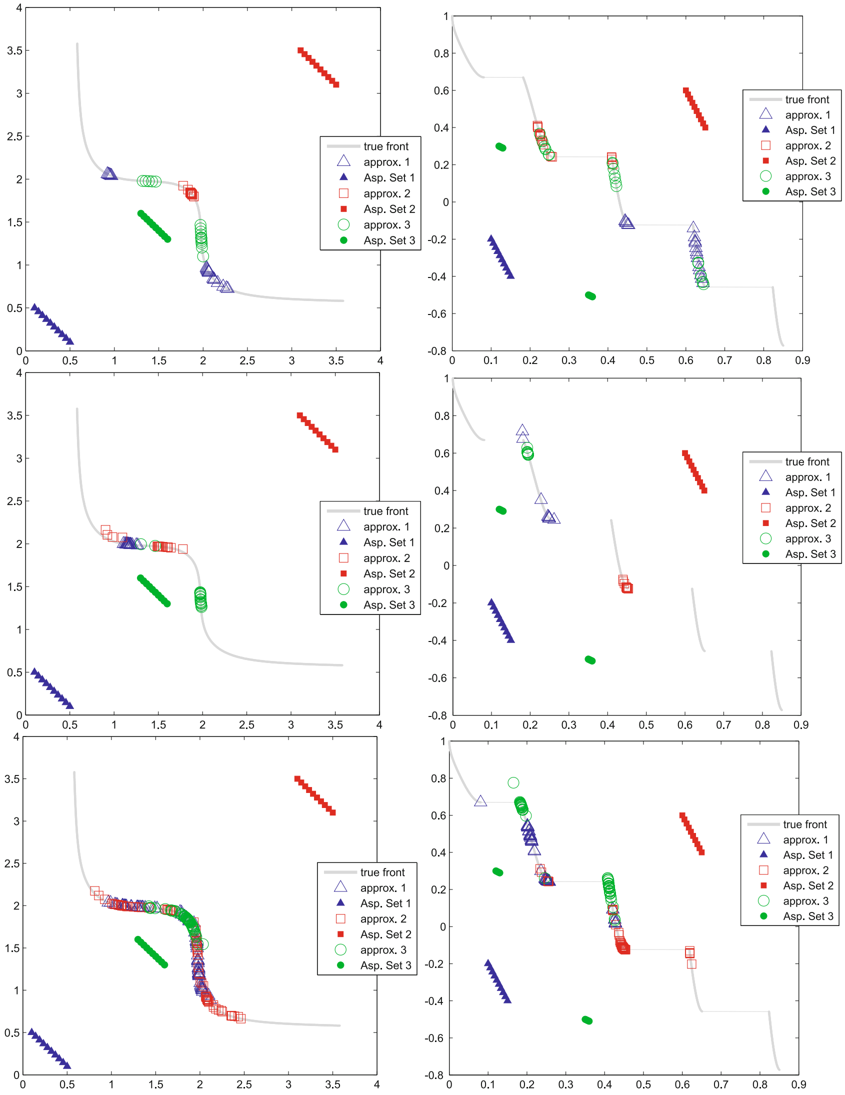


Fig. 3. Results of AS-EMOA (top), R-NSGA2 (middle) and R-NSGA2(M) (bottom) for DENT (left) and ZDT3 (right) using various reference sets and $p = 10$, $\mu = 20$

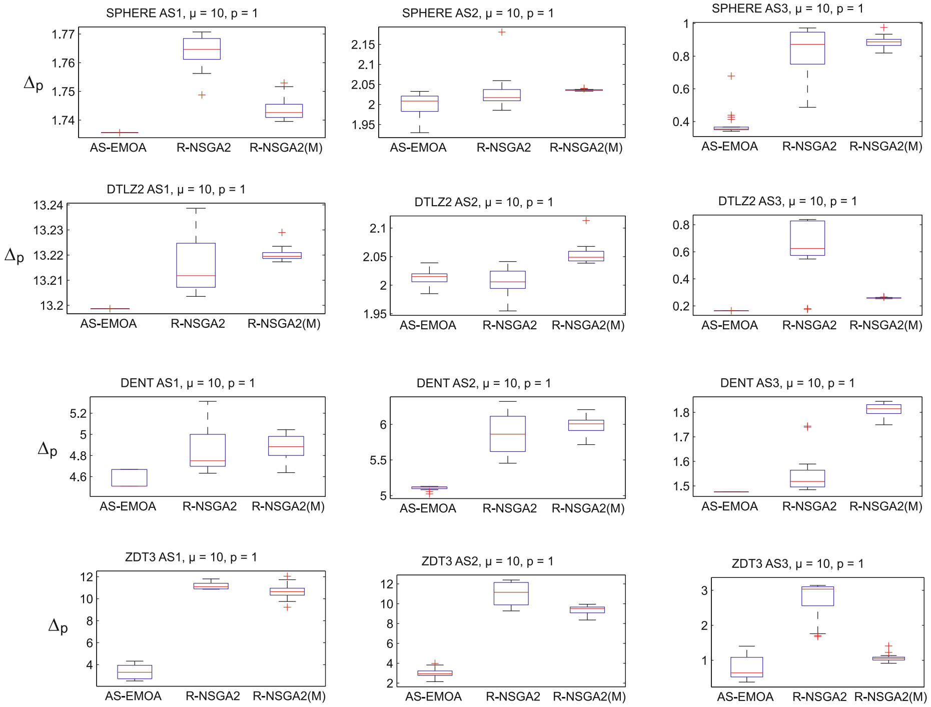


Fig. 4. Boxplots of Δ_p-values w.r.t. the aspiration set at final population, $p = 1$

The AS-EMOA was parametrized with SBX crossover ($p_x = 0.9$) and polynomial mutation ($p_m = 1/n$) and evaluated for $p \in \{1, 2, 5, 10\}$. Both R-NSGA2 variants used the same crossover and mutation settings as AS-EMOA with an additional $\epsilon = 0.001$ parameter for diversification as a suggested setting in [15].

5.2 Results

Figures 2 and 3 show the algorithm results. As no substantial differences could be detected for varying p and μ, results are given for exemplary settings. As a general observation, aspiration sets behind the true Pareto front do not hinder the algorithms from proceeding further to the front and generating good approximations.

Focussing SPHERE and DTLZ2 in Figure 2, it becomes obvious that AS-EMOA successfully approximates a region on the Pareto front as close as possible to the aspiration set. Additionally, the structure and extent of the aspiration set is reflected very well. This is in contrast to R-NSGA2, which tends to produce more concentrated approximation sets than the AS-EMOA. This is due to R-NSGA2 internals as the most important reference point within the aspiration set is the point closest to the true Pareto front and R-NSGA2 tries to concentrate the solutions in the vicinity of this point. The same tendency becomes obvious for DENT, see Figure 3. For ZDT3, found result sets are even shifted upwards or downwards in relation to the approximation set (AS1, AS2). For AS3, the lower part of the aspiration set is not approximated at all.

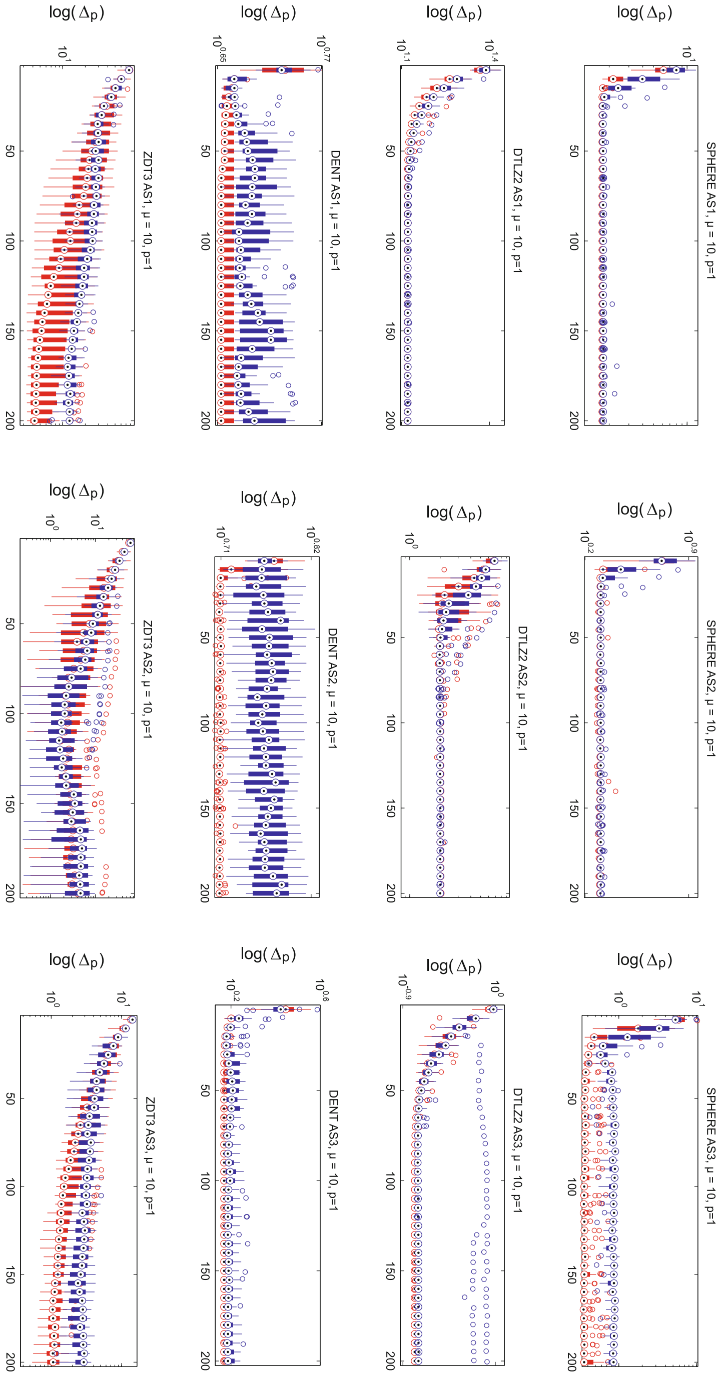


Fig. 5. Comparison of AS-EMOA (red) and R-NSGA2 (blue): Boxplots of $\varDelta_p$-values w.r.t. the aspiration set in the course of the generations, $p = 1$

The behavior of R-NSGA2(M) compared to the other algorithms depends on the structure of the aspiration sets. While similar behavior to AS-EMOA becomes visible for SPHERE (AS1) or DTLZ2 (AS2), shifted approximation sets are generated for the disconnected aspiration sets. This is also true for ZDT3 so that the lower part of the front is not focussed. Substantially different characteristics compared to AS-EMOA and R-NSGA2 are revealed for concave parts of the Pareto front combined with aspiration sets which are placed inside the convex hull of the concave part. R-NSGA2(M) is the only algorithm which focusses the central part of the front and does not generate two disconnected approximation sections. Which behavior is preferred in practice may depend on the specific problem at hand.

In terms of approximation quality measured by Δ_p, the AS-EMOA clearly outperforms R-NSGA2 and R-NSGA2(M) (see Figure 4). Only one setting (DTLZ2, AS2) does not show significant differences between AS-EMOA and R-NSGA2(M). The ranking between R-NSGA2 and R-NSGA2(M) coincides with some of the observations stated before: while R-NSGA2(M) exhibits the same shift between aspiration set and approximated solutions as R-NSGA2 (ref. Fig. 2, bottom left), the structure of aspiration sets is better preserved by R-NSGA2(M) (ref. Fig 2, bottom right). The property of R-NSGA2(M) approximating concave parts of the Pareto front leads to a significantly worse quality statement based on Δ_p compared to R-NSGA and AS-EMOA (ref. Fig. 4, DENT AS3). Still, no clear superiority of R-NSGA2(M) over R-NSGA2 can be discovered with respect to Δ_p.

Figure 5 shows the convergence behavior of AS-EMOA and R-NSGA2, reflected by the Δ_p values along the generations. In most cases AS-EMOA shows faster improvements while differences are highest and mostly significant for DENT and ZDT3. The overall tendency coincides with these observations. Furthermore, the fluctuation over the generations is much higher for R-NSGA2. A few settings, such as DTLZ2 combined with AS3, even show a stagnation at early generations. Note that R-NSGA2(M) was excluded from this comparison: The total convergence behavior of the parallel acting instances of R-NSGA2(M) becomes artificial and misleading by fluctuations due to intermediate aggregation and dominance filtering of multiple populations every generation.

6 Conclusions and Outlook

In this paper, we proposed an EMOA which is able to approximate favored parts of the Pareto front based on a given aspiration set of reference points. The distance measure used in this context is the averaged Hausdorff distance. Within our experiments the AS-EMOA successfully approximated the aspiration sets for different front shapes and aspiration sets in 2D objective space. It was demonstrated that even suboptimal aspiration sets do not hinder the AS-EMOA from reaching the desired region of the true Pareto front. We also showed that—with respect to Δ_p—AS-EMOA is superior to state-of-the-art approaches like R-NSGA2 and a modified variant based on concurrently working R-NSGA2 instances, so-called R-NSGA2(M). Moreover, AS-EMOA preserves the structure of the desired aspiration set within the approximated region on the Pareto front, while R-NSGA2–originally designed for a single reference point—tends to aggregate solutions close to the point of smallest distance to the Pareto front. Although R-NSGA2(M) is able to deal with an aspiration set and exposes interesting properties by approximating even concave parts of the Pareto front, parallel execution of multiple instances increases the number of function evaluations for this approach significantly. Further, AS-EMOA is superior to R-NSGA(M) in most cases regarding approximation quality with respect to averaged Hausdorff distance.

Overall, AS-EMOA shows promising perspectives for approximating subsets of the Pareto front based on sets of reference points. As next step we will extend this concept to higher dimensions as well; a suitable normalization within the Δ_p update procedure is a matter of further research. For future research directions it may be considered to also apply this approach in interactive optimization tasks.

Acknowledgements. Heike Trautmann and Christian Grimme acknowledge support by the European Center of Information Systems (ERCIS) and together with Günter Rudolph support by DFG project no. TR 891/7-1. Oliver Schütze acknowledges support from Conacyt project no. 128554. This work was also supported by the grant ANR-12-MONU-0009 (NumBBO) of the French National Research Agency and the DAAD project 57065955 ”Hybridization of indicator-based metaheuristics with modern local search methods in multiobjective optimization”.

References

[1] Wierzbicki, A.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Objective Decision Making, Theory and Application, pp. 468–486. Springer (1980)

[2] 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 16(4), 504–522 (2012)

[3] Rudolph, G., Schütze, O., Grimme, C., Trautmann, H.: An aspiration set EMOA based on averaged Hausdorff distances. In: Proceedings of the 8th Int'l. Conference on Learning and Intelligent Optimization (LION 8). Springer (to appear, 2014)

[4] Gerstl, K., Rudolph, G., Schütze, O., Trautmann, H.: Finding evenly spaced fronts for multiobjective control via averaging Hausdorff-measure. In: Proceedings of 8th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1–6. IEEE Press (2011)

[5] Trautmann, H., Rudolph, G., Dominguez-Medina, C., Schütze, O.: Finding evenly spaced pareto fronts for three-objective optimization problems. In: Schütze, O., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II. AISC, vol. 175, pp. 89–105. Springer, Heidelberg (2013)

[6] Rudolph, G., Trautmann, H., Sengupta, S., Schütze, O.: Evenly spaced pareto front approximations for tricriteria problems based on triangulation. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 443–458. Springer, Heidelberg (2013)

[7] Dominguez-Medina, C., Rudolph, G., Schütze, O., Trautmann, H.: Evenly spaced pareto fronts of quad-objective problems using PSA partitioning technique. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation (CEC 2013), Piscataway (NJ), pp. 3190–3197. IEEE Press (2013)

[8] Ignizio, J.: Goal programming and extensions. Lexington books. Lexington Books (1976)

[9] Branke, J.: Consideration of partial user preferences in evolutionary multiobjective optimization. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 157–178. Springer, Heidelberg (2008)

[10] Trautmann, H., Wagner, T., Biermann, D., Weihs, C.: Indicator-based selection in evolutionary multiobjective optimization algorithms based on the desirability index. Journal of Multi-Criteria Decision Analysis, 319–337 (2013)

[11] Deb, K., Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO 2006), pp. 635–642. ACM Press (2006)

[12] Figueira, J., Liefooghe, A., Talbi, E.G., Wierzbicki, A.: A parallel multiple reference point approach for multi-objective optimization. European Journal of Operational Research 205(2), 390–400 (2010)

[13] Heinonen, J.: Lectures on Analysis on Metric Spaces. Springer New York (2001)

[14] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007) ISBN 978-0-387-33254-3

[15] 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)

[16] Schütze, O., Laumanns, M., Tantar, E., Coello Coello, C.A., Talbi, E.: Computing gap free pareto front approximations with stochastic search algorithms. Evol. Comput. 18(1), 65–96 (2010)

Robust Optimization with Tchebysheff Decomposition for Mars Entry Probe Design

Liqiang Hou[1,2], Yuanli Cai[2], Jisheng Li[2], and Jin Liu[2]

[1] State Key Laboratory of Astronautic Dynamics, Xi'an Satellite Control Center,
Xi'an, 710043, China
houliqiang2008@139c.om,
[2] Xi'an Jiaotong University, Xi'an, Shannxi Province 710049, China

Abstract. An evidence based robust design optimization method with Tchebycheff decomposition is proposed for micro Mars probe design under epistemic uncertainty. Super-formula based super-ellipse is used for the probe geometric configuration instead of the conventional sphere-cone configuration. Evidence based multi-objective optimization(MOO) method is used to optimally design the probe. The MOO problem is casted into a set of scalar optimization problems with Tchebycheff decomposition. Individuals are grouped with an adaptive clustering algorithm. In each cluster,individuals are analyzed with Proper Orthogonal Decomposition(POD) technique, and sorted by the "energy" levels occupied by the components. A new population is produced by projecting the cluster centroid to the principal component vectors, modeling the distribution and reproducing new individuals. A strategy similar to steepest descend method in single-objective optimization is implemented for reproducing the new population, pushing forward the front to the true Pareto front. Performance and efficiency of the new algorithm are tested on a set of standard benchmark test problems. To reduce computational cost of evidence computation, an Evolutionary Binary Tree (EBT) algorithm and response surface model is employed. Finally, numerical simulation of a Mars micro probe heat shield design with the proposed optimization algorithm is presented.

Keywords: robust optimization, Tchebysheff decomposition, Mars entry probe.

1 Introduction

In this paper, an evidence based robust optimization method for a new type of Mars entry probe design is presented. Previous work primarily analyzes the 25-degree spherical-segment from Project Apollo and the 70-degree spherically-blunted cone from the Mars Viking, Pathfinder, and Exploration Rover missions [1] [2]. Both designs are spherical and have been applied over the past forty years. Some recent works introduced non-spherical geometry configuration in Mars and Earth entry probe deign. Unlike the conventional sphere-cone entry probes, new probe geometry is defined with a set of super-formula based super-ellipse [3]. Equation of the new probe geometry is a more generalized equation that can transform a polygon into an ellipse and then into a rounded-edge concave polygon. A wide, continuous range of cross sections of the entry probe geometry can then be generated by adjusting the equation parameters.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_19, © Springer International Publishing Switzerland 2014

With evidence based robust optimization strategy, both the epistemic and aleatory uncertainties from a poor or incomplete knowledge of the design parameters can be modeled. Values of uncertainties can be expressed as interval and associated Basic Belief Assignment or BPA values. Some applications of Evidence Theory to robust optimal design of space systems and space trajectories are given in recent works [4] [5] [6]. In these works, evidence based uncertainties in the design space and uncertain space were modeled and optimized through Optimization Under Uncertainties (OUU). Belief function was optimized (maximized) together with all the other criteria that define the optimality of the system.

Robust optimization of Mars micro probe design under epistemic uncertainty impacts can be formulated as OUU as well. Uncertainties of the entry probe design are from atmospheric model, aerodynamic model, and entry parameters. Due to lack of knowledge, accurate statistic distribution model for these parameters can not be given in preliminary design phase. Therefore, with evidence based optimization, objectives of the robust design are to minimize the TPS size, or interior temperature of the TPS, while at same time maximize its belief values in the context of epistemic model.

In this paper, a new MOO algorithm with Tchebysheff decomposition is performed for the robust design. Population of candidate solutions are adaptively clustered into groups. In each cluster, Proper Orthogonal Decomposition(POD) are performed to model the individual distribution, and decompose the solution space into a set of principle components. Centroid of the cluster are projected into the principle components and new population are then generated by exploring the subspace spanned by the principle components. A strategy similar to the steepest descent algorithm in single-objective optimization is implemented for determining searching direction and pushing the solution front to the true Pareto front.

During the optimization, computation of Belief and Plausibility grows exponentially with the number of dimensions of the uncertain space, and becomes intractable even for problems of moderate size. An Evolutionary Binary Tree (EBT) method is implemented for computing the belief and plausibility values of the epistemic uncertainty impacts [7]. Approximate computations of the believes and possibilities are integrated into the OUU. In the uncertainty subintervals, quadratic response surface methodology is employed to reduce the computational cost of evidence computation.

The remainder of the paper is structured as follows: Section 2 introduces models for the Micro Mars probe design, including the heat shield geometry and TPS sizing model. Section 3 describes the evidence based robust optimization problem of the Mars probe design. Section 4 presents POD assisted MOO with Tchebysheff decomposition. Numerical simulation results of the Mars entry probe design are given in Section 5, and Section 4 concludes this paper.

2 Micro Mars Probe Design

2.1 Heat Shield Geometry

Mathematical definition of the heat shield geometry is defined through the superellipse generated by superformula. Equations for the probe cross section of $r(\phi)$ with $(0 \leq \phi \leq 2\pi)$ is given by [3] [8]

$$r(\phi) = [|\frac{\cos(\frac{1}{4}m_1\phi)}{v_1}|^{n_2} + |\frac{\sin(\frac{1}{4}m_1\phi)}{v_2}|^{n_2}]^{-\frac{1}{n_1}} \tag{1}$$

where m_1 is the number of the the polygon sides, $v_1 = v_2 = 1$, n_1 and n_2 are modifiers transforming the polygon into ellipse ($n_2 = 2$) and rounded-edge concave polygon($n_2 > 2$). Fig.1 shows examples of cross section generated by the equation. Corresponding parameters of n_1,n_2 are listed in the figure as well. In this work a probe heat shield configuration with $m_1 = 4$ is studied. Axial profile of the probe is spherically-blunted cone. The heat shield geometry is then generated by sweeping 360^0 about the body axis using Eq.(1).

The cross section of the probe afterbody is defined using Eq.(1) as well, while its axial profile consists of spherical segment. Fig.2 shows an example of the heat shield geometry with $R_n = 0.05m$, $R_b = 0.4m$ and $\theta = 60^0$,$n_1 = n_2 = 1.75$, where R_n is radius of the nose, R_b is base radius and θ is semi-apex angle of the sphere cone respectively.

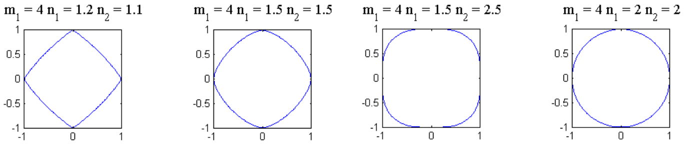


Fig. 1. Examples of base cross section

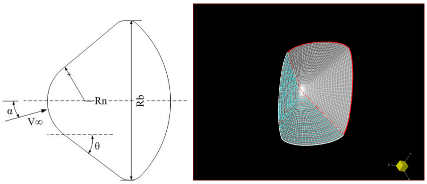


Fig. 2. Heat shield geometry

Suppose that the free stream Mach number $M_\infty \gg 1$, following the Modified Newtonian flow formula, the pressure coefficient on the heat shield surface can be computed as :

$$C_p = C_{p,t2}(\bar{\mathbf{v}}_\infty \cdot \hat{\mathbf{n}}/\bar{v}_\infty^2) \tag{2}$$

for $\bar{\mathbf{v}}_\infty \cdot \hat{\mathbf{n}} < 0$ and $C_p = 0$ for $\bar{\mathbf{v}}_\infty \cdot \hat{\mathbf{n}} \geq 0$, which represents the shadowed region. The relation between the stagnation pressure coefficient and free stream Mach number M_∞ is given by

$$C_{pt2} = \frac{2}{\gamma}(\frac{\gamma+1}{2})\frac{\gamma}{\gamma-1}(\frac{\gamma+1}{2\gamma - \frac{\gamma-1}{M_\infty^2}})^{\frac{1}{\gamma-1}} - \frac{2}{\gamma M_\infty^2} \tag{3}$$

where γ is specific heat rate.

Integrating the pressure coefficients upon the surface one can obtain the drag and lift coefficients, C_D and C_L [9] [10]. A program written in Matlab is designed for this purpose. In this program, inputs are the surface mesh, angle of attack, free stream Mach number, ratio of specific heats, and normalized length and area. Axial/normal and lift/drag force coefficients and the pitching moment coefficient are outputs. Structured surface mesh are created through commercial CFD software Numeca, save as Plot3d format. File of the surface meshes in Plot3d format is generated through the commercial software Numeca IGG.The aerodynamics estimator is acceptable for determining blunt-body shape hypersonic aerodynamics at fine mesh sizes with quite low run times (usually a fraction of a second).

2.2 TPS Sizing Model

The heat flux at stagnation point can be computed as [9]

$$\dot{q} = 1.89^{-8}\sqrt{\frac{\rho}{R_n}}v^3 \tag{4}$$

where $\rho(h)$ is atmospheric density, v is velocity and R_n indicates nose radius respectively.

Trajectory data of velocity v and h of the probe can be obtained through the ODE type trajectory model [10]. State of the dynamic model consists of $[r, \lambda, \phi, v, \vartheta, \xi]$, where r is radius, λ indicates longtitude, ϕ is latitude, ϑ is flight path angle, and ξ is azimuth angle respectively. Given $[r_0, \lambda_0, \phi_0, v_0, \vartheta_0, \xi_0]$ at entry point, trajectory data can be obtained by integrating the ODE equations. [10]

With surface radiation equilibrium condition, the relation of TPS surface temperature T_s and heat flux $\dot{q}$ is

$$\dot{q} = \varepsilon\sigma T_s^4 \tag{5}$$

where ε is emissivity factor, and $\sigma = 5.6703e - 8$ is the Stefan-Boltzman constant respectively.

For one-dimensional heat conduction (suppose that the heat transferred directly into TPS) , by neglecting the effects of decomposition, and pyrolysis gas flow , by using the semi-inifinite solid approximation, closed-form analytical solutions to the in-depth heat transfer equation can be obtained. The temperature at a depth x within the solid at time t is given by

$$T(x,t) = erf(\frac{x - \dot{x}t}{2\sqrt{\alpha t}})(T_0 - T_s) + T_s \tag{6}$$

where erf is the gaussian error function, $\dot{x}$ is TPS surface recession rate and T_0 is the initial temperature respectively. The thermal diffusivity α in Eq.(6) is given by

$$\alpha = \frac{k}{\rho c_p} \tag{7}$$

where k is thermal conductivity, $c_p$is specific heat, and ρ is density respectively. The equations above is for a simplified model for conceptual design.

Given trajectory parameters of the entry point, probe geometry configuration and TPS characteristic parameters, the interior temperature of the TPS can be obtained through Eq.(2) to Eq.(7).

3 Evidence Based Robust Design Optimization

3.1 Belief and Plausibility

Consider a system with interval type epistemic uncertainties, with Dempster-Shafer theory of evidence, evidence based robust optimization problem can be formulated as [4] [6]

$$\max_{v\in\Re,\mathbf{d}\in\mathbf{D},\mathbf{u}\in\mathbf{U}} Bel(f(\mathbf{d},\mathbf{u})<v);\ \min_{v\in\Re,\mathbf{d}\in\mathbf{D}} v \tag{8}$$

where v is the threshold to be minimized, $f(\mathbf{d},\mathbf{u})$ is the objective to be minimized in the design space $\mathbf{D}$ and uncertainty space $\mathbf{U}$. In this work, objective of the robust design is the interior temperature of the TPS.

Given uncertainty space $\mathbf{U}$, based on the information available concerning uncertain quantities, basic probability assignment (BPA) can be defined as

$$\sum_{\theta\in\mathbf{U}} m(\theta)=1; m(\theta)>0\forall\theta\in\mathbf{U}; m(\emptyset)=0; \tag{9}$$

When more than one parameter is uncertain, the focal elements are the result of the Cartesian product of all the intervals associated to each uncertain parameter. The BPA of a given focal element is then the product of the BPA of each interval. Given the uncertainty interval structure and BPAs, Belief and Plausibility of proposition A can be evaluated as

$$Bel(A)=\sum_{\forall\theta_i\subseteq A} m(\theta_i); Pl(A)=\sum_{\forall\theta_i\bigcap A\neq 0} m(\theta_i) \tag{10}$$

and have the relation

$$Bel(A)=1-Pl(-A) \tag{11}$$

3.2 Approximation of the Evidence Computation

An Evolutionary Binary Tree technique is performed for computing belief and plausibility of the performance $Bel(v))$ and $Pl(v)$ given the specific design vector **d** and uncertainty box **u** [7]. A brief description of the algorithm is given below:

First, given minimum and maximum values of performance parameter v, v_{min} and v_{max}, uniformly divide the interval v_{min}, v_{max} into n_v subintervals with $v_{j=1,\ldots,n_v}$. For each v_j with initial values $Bel(v_j)=0, Pl(v_j)=0$ do the following recursive binary partitioning:

1. Given uncertainty box $\mathbf{u} \in \mathbf{U}$, and design vector $\bar{\mathbf{x}}$, search the minimum and maximum performance values

$$v_{min} = \min_{\mathbf{u} \in \mathbf{U}} J(\bar{\mathbf{x}}, \mathbf{u}); v_{max} = \max_{\mathbf{u} \in \mathbf{U}} J(\bar{\mathbf{x}}, \mathbf{u}) \tag{12}$$

2. Partition the uncertain box $\mathbf{u}$ equally into two boxes $\mathbf{u}^l$ and $\mathbf{u}^r$
3. Given box $\mathbf{u}^l$, search v^l_{min} and v^l_{max} as Eq.(12) shown.
if $v^l_{max} \leq v_j$

$$Bel(v_j) = Bel(v_j) + BPA(\mathbf{u}^l)\ ,\ Pl(v_j) = Pl(v_j) + BPA(\mathbf{u}^l)$$

else if $v^l_{min} < v^l_{max}$, partition u^l into two new boxes, and remove u^l;
For each $\mathbf{u}^m$ with $v^m_{min} < v_j < v^m_{max}$

$$Pl(v_j) = Pl(v_j) + BPA(\mathbf{u}^m)$$

Repeat the steps till the termination condition is met, e.g. maximum number of partitions is achieved or the BPA value of the subintervals is below a specified threshold.

Note that in this algorithm, the most computational cost comes from searching v_{min} and v_{max} in the uncertain boxes, and number of the uncertain boxes needed to be explored increases exponentially with the uncertain parameters. Some approximation techniques are therefore employed in this work to reduce the cost for searching v_{min} and v_{max} .

The approximation is conducted through adaptive sampling and response surface model. In each uncertain box $\bar{\mathbf{u}}$, samples are generated uniformly. Sample size is set to be proportional to the product of BPA and longest distance of the box. With quadratic response surface model, the fitted second-order model in matrix form is as follows:

$$\hat{y} = \hat{\beta}_0 + x'\mathbf{b} + x'\mathbf{B}x \tag{13}$$

Coefficients **b** and **B** can be obtained through nonlinear regression program. Corresponding estimated response value at the stationary point can be computed as

$$\hat{y_s} = \hat{\beta}_0 + \frac{1}{2}x_s^{'}\mathbf{b} \tag{14}$$

where the stationary point $x_s = -\frac{1}{2}\mathbf{B}^{-1}\mathbf{b}$. Estimated response values can then be put into the evidence belief computation instead of v_{min} and v_{max} which cost quite higher computational resources.

4 POD Assisted MOO with Tchebysheff Decomposition

4.1 Chebyshev Decomposition

Consider a MOO problem

$$\min F(x) = [f_1(\mathbf{x}), ..., f_m(\mathbf{x})]^T \tag{15}$$
$$\text{subject to } \mathbf{x} \in \mathbf{D}$$

where Ω is decision variable space, $F : \mathbf{D} \to \Re^m$ consists of m real-valued objective functions. With Tchebysheff decomposition, the MOO equation can be reformulated as [11] [12] [13]

$$\min_{\mathbf{x} \in \mathbf{D}} g(f(\mathbf{x}), \lambda, \mathbf{z}) = \min_{\mathbf{x} \in \mathbf{D}} \max_{l=1,...,m} \{\lambda_l |f_l(\mathbf{x}) - z_l|\} \tag{16}$$

where $\mathbf{z} = [z_1, ..., z_m]$ is the reference point

$$z_l = \min_{\mathbf{x} \in \mathbf{D}} f_l(\mathbf{x}) (l = 1, ..., m) \tag{17}$$

and$\lambda_{l=1,..,m}$ is the weight vector. With Eq.(16) and Eq.(17), and different wight vectors, different Pareto optimal solutions can be obtained.

4.2 Proper Orthogonal Decomposition

With Tchebysheff decomposition, individuals with better merit can be selected. In this section, a new MOO evolution strategy based on Proper Orthogonal Decomposition(POD), also named Principle Component Analysis (PCA) techniques is proposed.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. Consider M snapshots $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M \in \Re^n$, construct the kernal matrix $\mathbf{K} \in \Re^{n \times n}$

$$K = \sum_{j=1}^{M} (\mathbf{x}_j - \mathbf{x}_c)(\mathbf{x}_j - \mathbf{x}_c)^T \tag{18}$$

where x_c is the center of snapshots $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_M$. Compute eigenvectors $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M]$ of K and corresponding eigenvalues with $\lambda_1 \geq \lambda_2 ..., \lambda_M$. Relative "energy" of the ith principle component can then be given by $\lambda_i / \sum_{j=1}^{M} \lambda_j$. Figure 3 shows PCA of a multivariate Gaussian distribution centered at (2,4,5) with covariance

$$\sigma = \begin{pmatrix} 1.0\ 0.2\ 0.7 \\ 0.2\ 1.0\ 0.4 \\ 0.7\ 0.4\ 1.0 \end{pmatrix} \tag{19}$$

Vectors shown in the picture are the eigenvectors of the covariance matrix scaled by the square root of the corresponding eigenvalue, and shifted so their tails are at the mean.

For multivariate dataset in a high-dimensional data space, PCA can supply the user with a lower-dimensional picture viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

With POD decompositions, orthogonal components spanning the data hyperplane can be obtained. The decomposition results reveal the internal structure of the data in a way that best explains the variance in the data. Note that both the components of highest and lowest "energy" have their own meanings. Since all the individuals in

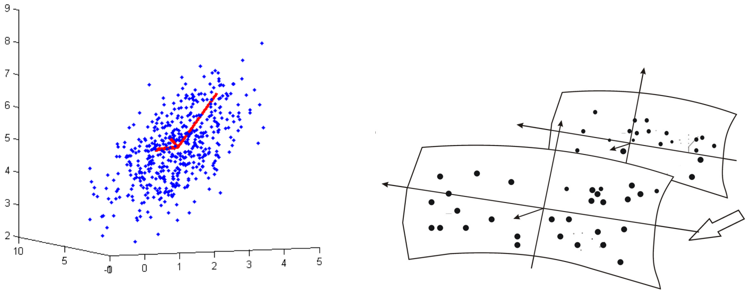


Fig. 3. PCA of a multivariate Gaussian distribution

the population can be described by principle components, and in the new coordinate system, the components with low "energies" are in the direction of short axes, while the higher ones are in the direction of the longer axes, if new population front with better merit are obtained, those components with lowest "energies" can then be treated as the "deepest" descend direction with better performance, like those problems in single objective optimization.

Therefore, one can use the high level energy components as basis for producing new front, project the population centroid to the longest axes in new coordinate system, and uniform randomly generate the individual component along these axes. Boundaries of the population along these axes are extended such that more possibilities could be generated for the optimizer to explore the design space. Similar things can be done for the those components with lowest "energies". Axes with the lowest "energies" normal to the hyperplane spanned by high level principle components could be used for generating individual components with better merits, such that the front could be pushed forward to the Pareto front. Once the searching direction is determined, one can use adaptive step or constant step to explore the design space. In this work, magnitude of the searching step of the principle component is set to be proportional to the ratio of the "energy" of the component occupies to the total "energies".

The projection of the centroid onto the principle component $\mathbf{v}_i$ can be computed as

$$\mathbf{x}_{ref_i} = \frac{\mathbf{x}_c \cdot \mathbf{v}_i}{|\mathbf{v}_i|^2}\mathbf{v}_i \tag{20}$$

Note that in this algorithm, only those principle components with highest and lowest energies are used for reproducing the new population. As mentioned above, PCA could supply the analyzer a lower dimensional informative view of the transformed data by using only the first few principal components. This could also help reduce computational cost during optimization. With more and more non-dominate individuals generated, population generated by principle components eventually converge to the true Pareto set.

The centroid of the population can be determined by cluster methods, e.g. k-mean cluster, PCA clustering, and affinity propagation clustering method etc [14]. In this work,the affinity propagation method is performed for adaptive grouping individuals. No predefined number of groups is required. Groups are determined by the messages sent and received between individuals [14].In each group, POD technique and Tchebysheff decomposition are used to generate new population.

4.3 Framework of the POD Assisted MOO

The algorithm works as follows:

Step1. Initialization

Step 1.1) Generate n_{pop} initial population $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_{n_{pop}}]$ randomly or by a problem-specific method.

Step 1.2) Initialize weight vector $\lambda = [\lambda_0, ..., \lambda_{n_{pop}-1}]^T$ using uniform spread method. For bi-objective case, the weight vector is

$$\lambda_{i=1,...,n_{pop}} = \left[\frac{i-1}{n_{pop}}, \frac{n_{pop}-i+1}{n_{pop}} \right]^T \tag{21}$$

Step 1.3) Initialize the reference vector $\mathbf{z}^* = [z_1, ..., z_m]^T$, where z_i is the best value found so far for the objective value f_i ;

Step2. Update population with POD decomposition

step 2.1) Adaptive group the individuals into groups;

step 2.2) For each group do: compute the principle components, and sort the eigenvectors with respect to the "energies" they occupy(see Eq.(18).

Select n_h components $\mathbf{v}_h$ with energy higher than specific level as reference basis for new population, project the cluster centroid to the eigenvectors $\mathbf{v}_h$, extend the boundary and generate n_{pop} individual components uniformly distributed in direction of $\mathbf{v}_h$

$$\mathbf{x}_h = \mathbf{x}_{ref_h} + \rho_h[\text{rand}(n_{pop}, 1) - 0.5]\mathbf{v}_h \tag{22}$$

with the step

$$\rho = \sigma \sqrt{\frac{|\lambda_i|}{\sum_{i=1,nx} \lambda_i}} \tag{23}$$

where σ is a constant between 1.2 and 1.8.

Select n_l components $\mathbf{v}_l$ with energy lower than specific level, generate n_{pop} individual components distributed uniformly in direction of $\mathbf{v}_l$ normal to $\mathbf{v}_h$ to push forward the front

$$\mathbf{x}_l = \mathbf{x}_{ref_l} + \rho_l[\text{rand}(n_{pop}, 1) - 0.5]\mathbf{v}_l \tag{24}$$

New individuals are then generated by adding all the components generated by both the basis and normal components.

$$\mathbf{x}_{new} = \mathbf{x}_h + \mathbf{x}_l \tag{25}$$

step 3) update non-dominated solutions. For $i = 1 : n_{pop}$ do:

Update reference point $\mathbf{z}^*$. For each $j = 1, ...m$ and $i = 1, ..., n_{pop}$, if $z_j < f_j(\mathbf{x}_{new_i})$, set $z_j = f_j(\mathbf{x}_{new_i})$. For each individual $\mathbf{y}_i \in \mathbf{x}_{new}, (i = 1, ..., n_{pop})$, if $g(\mathbf{y}_j|\lambda, \mathbf{z}) < g(x_j|\lambda, \mathbf{z})$, set $\mathbf{x}_j = \mathbf{y}_j$. If no solutions can be found, or the solutions fall into the trap, e.g. size of the index set $i_{new} = find(g_{new} \prec g_{old})$ is equal to n_{pop}, re-initialize the population randomly and go back to step 1 till effective candidates are obtained.

Step 4)

If stopping criteria is met, e.g. the maximum iteration number, stop the iteration, otherwise, goto step2 .

4.4 Experimental Results

To test the new MOO algorithm's performance, comparison of NSGA-II and the proposed algorithms is performed on a set of high dimensional MOO benchmark problems: ZDT1 ,ZDT2, ZDT3 and ZDT6 [15] [16] [17].

Only 4 high level and 3 low level principle components are used for generating the populations, i.e. $n_h = 4$ and $n_l = 3$. The constant σ in Eq.(23) is set to 1.6. Fig.4 shows the new MOO algorithm solutions after 20 iterations. Corresponding solutions of NSGA-II after 20 generations are listed in the figure as well. Both population size of the new MOO algorithm and NSGA-II are set to 100. Crossover probability and and mutation probability of NSGA-II are set to $p_c = 0.9$ and $pm = 1/n$, where n is the number of decision variables. The distribution indices in both SBX and the polynomial mutation in NSGA-II are set to be 20.

It appears that with new MOO, significant improvement of the efficiency can be achieved. Solutions of the new MOO algorithm are quite close to the true Pareto set, while for the counterpart algorithm, NSGA-II, some more iterations (the number maybe up to several dozens) are required before the solutions converge to the Pareto set.

Table 1. Numbers of function evaluations after 20 iterations

Numbers of function evaluations	NSGA-II	POD assisted MOO
ZDT1	2013 $\sim$ 2020	2000
ZDT2	1987 $\sim$ 2009	2000
ZDT3	1996 $\sim$ 2010	2000
ZDT6	1982 $\sim$ 2009	2000

Table 1 shows the numbers of the function evaluations. It appears that with same number of iterations, the number of function evaluations of POD assisted MOO algorithm is similar to that of NSGA-II. Note that in Table 1 , cost of finding a good initial population of the POD assisted MOO algorithm is not taken into account. Once a good initial population is obtained, the algorithm will converge quickly to the Pareto set. The new algorithm may take several iterations (in some cases the number may be up to tens)to find a good initial population (see re-initialize population in step 3). Considering that it may take several dozens of iterations for the algorithms with genetic operations, e.g. NSGA-II and MOEA/D, to converge to the Pareto set, therefore, the total cost of the new algorithm is still lower than the cost of NSGA-II and MOEA/D. Figure 4 illustrates the differences between POD-MOO and NSGA-II.

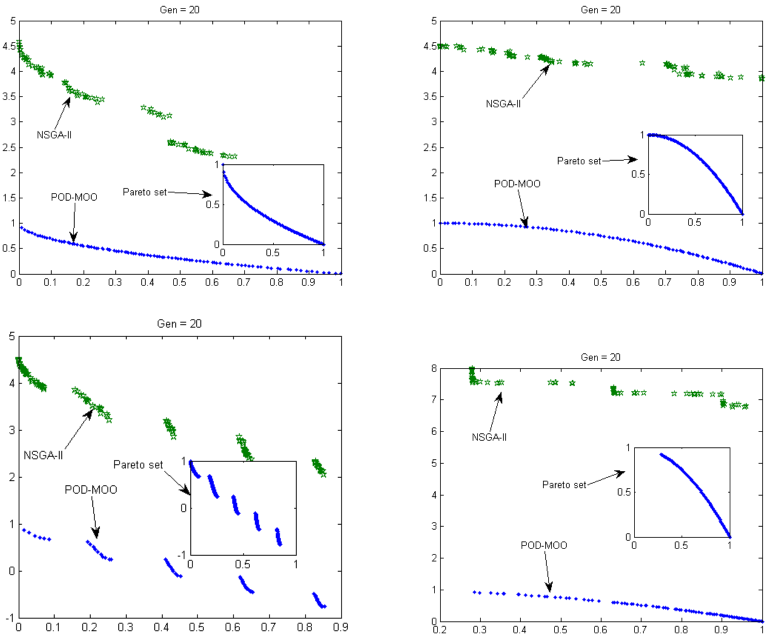


Fig. 4. Pareto set after 20 iterations of the new MOO (ZDT1,ZDT2,ZDT3,ZDT6)

5 Numerical Simulation

In this section, the robust optimization of the new micro entry probe design is presented. The radius of cross section of the probe R_b is no more than 0.4m . Initial entry trajectory parameters of radius r_0, longitude λ_0, latitude ϕ_0, velocity v_0,flight path angle ϑ_0 and velocity azimuth angle ξ_0 are from mission MER-B in 2004. Table 2 shows their values. Note that in these parameters, the flight path angle is closely related to heat load and overload during the entry. A too shallow flight path angle will make the probe bounce off the atmosphere. If it is too steep, the probe will descend into dense atmosphere layers too quickly and incinerate.

The design space consist of the heat shield parameters of semi apex angle θ, nose radius R_n, super formula factors n_1 , and n_2. Table 3 shows lower and upper bounds of the design variables. Initial TPS thickness is $L_{TPS} = 1.1cm$ and mass of the probe is set to $m_0 = 12kg$. The heat shield layer is made of SLA-561V, an ablative type TPS material widely used in planetary entry probe sent by NASA. Material properties of SLA-561V in the simulation are from TPSX database [18]. Table 4 shows BPA structure of evidence related epistemic uncertainties of ϑ_0, v_0, m_0, ρ, C_D and C_L etc.

The new developed robust MOO algorithm is applied to the design optimization of the probe. Figure 5 shows robust Pareto solutions after 50 iterations. Both belief and plausibility levels of the optimized TPS performance are listed in the figure. Two optimal solutions, solution A with belief of 0.878, and solution B with belief of 0.318 are

Table 2. Initial Entry Condition

Parameter	r_0	λ_0	ϕ_0	v_0	ϑ_0	ξ_0
Value	3392.3	161.776	-17.742	5.628	-11.495	79.025
Unit	km	(^{0})	(^{0})	(^{0})	(^{0})	(^{0})

Table 3. Design space

Parameter	Lower Bound	Upper bound	Unit
R_n	0.04	0.20	(m)
θ	35	75	(^{0})
n_1	1.10	2.0	
n_2	1.10	2.0	

Table 4. BPA structure of uncertainties

Parameter	Value	Lower Bound	Upper Bound	BPA
ϑ_0	-11.495^0	-5%	-2%	0.35
		-2%	0	0.25
		0	2%	0.30
		2%	3%	0.10
v_0	5.628 km/s	-5%	-3%	0.45
		-3%	0	0.25
		0	2%	0.20
		2%	3%	0.10
m_0	12kg	-10%	-5%	0.35
		-5%	0	0.25
		0	5%	0.20
		5%	10%	0.20
ρ	$\rho(h)$	-10%	-5%	0.05
		-5%	0	0.25
		0	5%	0.30
		5%	10%	0.40
C_D	$C_D(Ma, R_n, R_b, \theta)$	-10%	-5%	0.05
		-5%	0	0.25
		0	5%	0.30
		5%	10%	0.40
C_L	$C_L(Ma, R_n, R_b, \theta)$	-10%	-5%	0.05
		-5%	0	0.25
		0	5%	0.30
		5%	10%	0.40

picked from the final Pareto optimal-set. The corresponding geometry configurations are shown in the figure as well. Note that solution A has an semi-apex angle close to the upper bound, also the adjusting parameters n_1 and n_2 are quite bigger than those of solution B. It appears that the uncertainty impacts of the probes with rounded edges

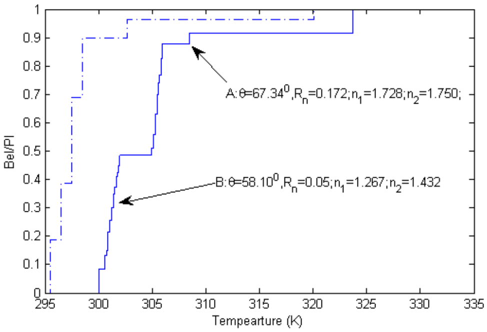


Fig. 5. Heat shield geometry solutions(A and B)

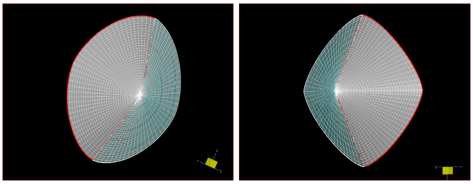


Fig. 6. Heat shield geometry solutions(A and B)

are less than those with sharp edges. In the case of solution B which has a smaller R_n, belief of a better temperature performance is quite low. This could be explained by the heat flux equation(see Eq.(4)). With this equation, heat flux increases as R_n becomes smaller.

6 Conclusion

In this paper, a PDO assisted robust optimization method with Tchebysheff decomposition for Mars micro entry probe design is presented. The heat shield geometry of the Probe is defined by a set of super-formula. Equations of the probe geometry could generate a wide, continuous range of cross sections. More geometric freedom can then be given for design optimization. Evidence type epistemic uncertainty impact analysis are integrated into the design optimization. Robust design is implemented through a new developed POD assisted MOO algorithm using Tchbysheff Decomposition. The statistical procedure with orthogonal transformation converts set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. A gradient-like searching strategy is then developed to search Pareto solutions. Efficiency of the new algorithm is experimentally demonstrated through a set of classical benchmark MOO test functions.

The new approach provides a way for entry vehicle TPS sizing optimization under epistemic uncertainties. One of the major problems in evidence based robust optimization is the computational cost of evidence computation. The new approach integrates quadratic response surface based approximations for evidence computation into the robust MOO. Computational cost is reduced by evidence approximation and new efficient MOO algorithm. In the new robust MOO algorithm, only the principle components with energies higher than or lower than specified levels are used to construct the solution space. This could also help reduce computational cost. A more in depth investigation of the robust design in sophisticated scenario, e.g. robust design with variable fidelity models is underway.

References

[1] Mitcheltree, R., Moss, J., Cheatwood, F., Greene, F., Braun, R.: Aerodynamics of the mars microprobe entry vehicles. Journal of Spacecraft and Rockets 36(3), 392–398 (1999)

[2] Wright, M.J., Tang, C.Y., Edquist, K.T., Hollis, B.R., Krasa, P., Campbell, C.A.: A review of aerothermal modeling for mars entry missions. AIAA Paper 443, 4–7 (2010)

[3] Johnson, J.E., Starkey, R.P., Lewis, M.J.: Aerodynamic stability of reentry heat shield shapes for a crew exploration vehicle. Journal of Spacecraft and Rockets 43(4), 721–730 (2006)

[4] Vasile, M.: Robust mission design through evidence theory and multiagent collaborative search. Annals of the New York Academy of Sciences 1065(1), 152–173 (2005)

[5] Vasile, M.L.: A behavioral-based meta-heuristic for robust global trajectory optimization. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 2056–2063. IEEE (2007)

[6] Zuiani, F., Vasile, M., Gibbings, A.: Evidence-based robust design of deflection actions for near earth objects. Celestial Mechanics and Dynamical Astronomy 114(1-2), 107–136 (2012)

[7] Vasile, M., MInisci, E., Wijnands, Q.: Approximated computation of belief functions for robust design optimization. In: 14th AIAA Non-Deterministic Approaches Conference (2012)

[8] Johnson, J.E., Starkey, R.P., Lewis, M.J.: Aerothermodynamic optimization of reentry heat shield shapes for a crew exploration vehicle. Journal of Spacecraft and Rockets 44(4), 849–859 (2007)

[9] Anderson, J.D.: Hypersonic and high temperature gas dynamics. Aiaa (2000)

[10] Reagan, F.J., Anandakrishnan, S.M.: Dynamics of atmospheric re-entry. Aiaa (1993)

[11] Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)

[12] Zuiani, F., Vasile, M.: Multi agent collaborative search based on tchebycheff decomposition. In: Computational Optimization and Applications, pp. 1–20 (2013)

[13] Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Transactions on Evolutionary Computation 14(3), 456–474 (2010)

[14] Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315(5814), 972–976 (2007)

[15] 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)

[16] Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A.K., Padmanabhan, D.: Reliability-based optimization using evolutionary algorithms. IEEE Transactions on Evolutionary Computation 13(5), 1054–1074 (2009)

[17] Mittal, S., Deb, K.: Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives. IEEE Transactions on Evolutionary Computation 13(3), 554–565 (2009)

[18] Squire, T.H., Milos, F.S., Hartlieb, G.C.: Aerospace materials property database (tpsx). Journal of Spacecraft and Rockets 46(3), 733–736 (2009)

Part VIII

Practical Aspects of Evolutionary Algorithms

Optimized Fourier Approximation Models for Estimating Monthly Streamflow in the Vanderkloof Dam, South Africa

Josiah Adeyemo and O. Oluwatosin Olofintoye*

Department of Civil Engineering and Surveying
Durban University of Technology
Durban, South Africa
olofintoye.oo@gmail.com

Abstract. Parametric probability distribution models and nonparametric approximation models were developed for estimating monthly streamflow in the Vanderkloof dam. The probability distribution functions include Normal, Log-normal, PearsonIII, Log-Pearson III, Gumbel and Log-Gumbel probability density functions while the nonparametric approach involves the development of monthly streamflow models using numerical harmonic Fourier series approximation procedure. The parameters of the Fourier series were optimized using differential evolution (DE) algorithm. For a comparison of nonparametric and parametric models, the Mean Relative Deviation (MRD) and Mean Square Relative Deviation (MSRD) statistics were used to measure the goodness-of-fit of the developed models while the Wilcoxon sign rank test was adopted to determine if there is a statistical significant difference in the performance of the models. Results show that the nonparametric Fourier approximation model estimates streamflow into the Vanderkloof reservoir better than the parametric methods. It is concluded that the Fourier series approximation may be adopted as an alternative approach for streamflow frequency analysis.

Keywords: Parametric, Nonparametric function, Fourier series approximation, Differential evolution, Vanderkloof dam.

1 Introduction

The quantity of water contained in surface streams at any given time, while small in terms of the earth's total supply, is of considerable importance to those engaged in water resources planning, development and management. A knowledge of the quantity and/or quality of streamflow is critical to such fields as municipal and industrial water supply, flood control, streamflow forecasting, reservoir design, management and operation, hydroelectric power generation, irrigation, water-based recreation, navigation, fish and wildlife management, drainage and water and wastewater treatment. In addition, the ability to provide reliable forecasts or estimates of flow for

* Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_20, © Springer International Publishing Switzerland 2014

periods in the future is of great value in operating storage, retention and other works and in planning proper actions during times of flood and low flows[1].

Less than 1% of the water of the earth is available as fresh water on land in streams and rivers. The rest is contained either in the oceans or in form of frozen ice on mountain tops and glaciers [2, 3]. Consequently, the existence of man and other living creatures in the various ecosystems of the earth is greatly dependent on the occurrence, distribution, amount, frequency and environmental impacts of this important hydrologic variable. Therefore, detailed analyses of streamflow events with the aim of devising ways by which future occurrences can be predicted or estimated within certain degrees of freedom is of paramount importance in hydrological studies [4].

Due to the complexities involved in flow estimation, much attempts have been made in order to understand the phenomena governing the streamflow process, so as to be able to predict future events and propound ways to mitigate against extreme events that may have catastrophic effects on the environment [5-7]. To this end, many approximation formulas have been developed in response to this need. The earliest of these were usually crude empirical statements, whereas the trend now is to develop descriptive equations based on physical processes[1]. Furthermore, several mathematical, probabilistic and numerical models have been used by researchers in the field of hydrology in attempts to estimate streamflow events within certain degree of accuracy [3, 8-11].

In the past, hydrologic processes were believed to come from a random distribution and could be interpreted or explained only in a probabilistic sense and probability distribution functions have been used to model various hydrologic processes [1]. Recent researches have however shown that against parametric methods, nonparametric functions may be employed as alternative approaches for hydrologic frequency analysis[3, 11]. For instance, Guo, [12] proposed a nonparametric variable kernel estimation model which provides an alternative way in flood quantile estimation when historical floods data are available and was shown that the nonparametric kernel estimator fitted the real data points closer than its parametric counterparts.

The Fourier series approximation, which uses an infinite sum of sine and cosine terms to approximate a function have been found useful in modeling hydrological processes. For example,[13] compared the performance of five parametric density functions and a nonparametric Fourier approximation function in estimating annual rainfall of Ilorin, Nigeria. The parametric density functions include Lognormal, Pearson III, Log-Pearson III, Gumbel, and Log-Gumbel probability distribution functions. The Fourier series was fitted using a numerical harmonic analysis suggested by Stroud, [14]. For comparison of nonparametric and parametric models, the Mean Relative Deviation (MRD) and Mean Square Relative Deviation (MSRD) statistics were used to measure the goodness-of-fit of the developed models. Results showed that the Fourier series approximation model predicts annual rainfall of Ilorin better than the parametric methods. It was concluded that the Fourier series approximation may be used as an alternative approach for hydrologic frequency analysis. Jou et al., [15] compared parametric density functions (Normal, two and three parameter Lognormal, two parameter Gamma, Pearson, Log-Pearson type III and Gumbel extreme value type I) with a nonparametric Fourier series to estimate annual precipitation for five rain gauge stations in Iran. The result showed that the Fourier series predict annual precipitation better than other parametric methods based on a Mean Relative Deviation (MRD) and Mean Square Relative Deviation (MSRD) goodness-of-fit test. It was concluded that Fourier series can be used as a better alternative approach for precipitation frequency analysis.

This study further investigates the performance of the nonparametric numerical Fourier series approximation in modeling hydrological processes by developing models to estimate monthly streamflows in the Vanderkloof dam, South Africa. Herein, a methodology in which an evolutionary optimization technique is employed to optimize the parameters of a numerical Fourier series approximation for hydrologic modeling is introduced.

2 Materials and Method

2.1 Study Area

Vanderkloof dam is maintained by the Department of Water Affairs (DWA), South Africa. It was constructed in 1973 and commission in 1977. It is situated near Petrusville in the Northern Cape province of South Africa on latitude 29.99222°S and longitude 24.73167°E, with a relief of about 1200 m above mean sea level(AMSL). Average yearly rainfall is between 400 mm and 200 mm while the mean annual temperature is around 18.9^{o} C. The reservoir is fed by the Orange River, South Africa's largest river which rises in the Lesotho Highlands on the eastern Drakensberg plateau at an altitude of about 3,300 m AMSL. Vanderkloof Dam is the second-largest dam in South Africa (in volume), having the highest dam wall in the country at 108 metres. It has a capacity of 3,187,557,000 cubic metres and a surface area of 133.43 square kilometers when full. The dam mainly serves to generate hydroelectricity and provide water for irrigation. It is also used for flood control [16].

2.2 Data Analysis

Monthly reservoir inflow data for the Vanderkloof dam was obtained from the Department of Water Affairs, Bloemfontein, South Africa. DWA is the agency responsible for the measurement, control and storage of hydrological data of major dams in South Africa. The nature of data is streamflow volume in mega liter (Ml) recorded for every month of the year. This was converted to mega cubic meter (Mm^3) for use in the analysis herein. A period spanning 36 years of data (1977 – 2013) was used for the analysis. A summary of the statistics of the monthly series is presented in Table 1. The series were ranked according to Weilbull's plotting position and the corresponding return periods were estimated. The series were evaluated using six methods of probability distribution functions, Normal, Log-Normal (LN), Pearson III, Log-Pearson type III (LP$_3$), Gumbel extreme value type1(EVI) and Log-Gumbel (LG). The monthly series were also fitted with a Fourier approximation model. Goodness-of-fit tests were adopted for the selection of the best fit models.

Table 1. Summary of statistics for monthly streamflow of Vanderkloof dam (1977 – 2013)

	Statitics					
Month	Mean value, (Mm^3)	Standard deviation, (Mm^3)	Skewness coefficient, G	Median (Mm^3)	Minimum (Mm^3)	Maximum (Mm^3)
Jan	539.29	802.57	4.64	321.28	104.44	4743.02
Feb	689.35	660.42	1.52	462.00	41.62	2716.39
Mar	771.99	1054.10	3.89	432.80	38.11	5987.26
Apr	488.49	444.21	1.33	319.17	22.69	1710.15
May	359.00	302.45	1.91	278.70	40.08	1329.35
Jun	294.29	228.02	2.87	249.64	20.86	1298.35
Jul	298.87	213.33	1.46	238.34	40.50	924.35
Aug	319.46	203.46	1.29	266.72	39.25	911.09
Sep	366.47	302.30	2.92	287.27	27.09	1712.45
Oct	385.10	335.40	2.98	298.62	53.83	1868.55
Nov	427.62	404.71	1.96	267.83	72.76	1823.81
Dec	469.37	417.93	3.10	358.59	45.08	2398.44

2.3 Evaluation of Probability Distribution Models

The frequency analysis was carried out in accordance with standard procedure [1, 10, 17, 18]. A description of the distribution and parameter estimation methods are not presented in this study because they are available in other publications. Therefore, only the Fourier series approach is described herein.

2.4 Development of Numerical Fourier Approximation Models

Fourier series models were developed for each month of the year following an N-point analysis procedure suggested by Stroud, [14] and Olofintoye and Adeyemo, [13]. N is the number of experimental data points employed in fitting the series. The Fourier series representation of a function is given by equation (1) below [14, 19].

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} \{a_n \cos(nx) + b_n \sin(nx)\} \tag{1}$$

where

$$a_0 = \frac{1}{\pi} \int_0^{2\pi} f(x)dx = 2 \times \text{mean value of f(x) over a period} \tag{2}$$

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos(nx)dx = 2\times \text{mean value of f(x) cos(nx) over a period} \quad (3)$$

$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin(nx)dx = 2\times \text{mean value of f(x) sin(nx) over a period} \quad (4)$$

The main task in developing a numerical Fourier approximation model is determining the coefficients a_0, a_1, a_2, … and b_1, b_2, b_3… of the series using equations (2), (3) and (4) so that the series accurately represent the function [14]. Furthermore, the number of data points to use and the number of sine and cosine terms to include in the series has to be determined. When the numbers of terms are insufficient, under fitting occurs while too many terms results in over fitting [19]. In this study, Differential evolution algorithm was employed to obtain the optimal number of terms required to give a best fit to the data points. Differential evolution (DE) developed by Storn and Price, [20] 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. Further Details on DE may be found in Price et al., [21]. The constrained optimization problem for fitting the Fourier approximation series using DE is stated as follows:

Minimize MRD
Subject to:
$12 \leq \textit{Number of Fit Points} \leq 100$
$0 \leq \textit{Number of Cosine Terms} \leq 36$
$0 \leq \textit{Number of Sine Terms} \leq 36$

where MRD is Mean Relative Deviation in equation 5. A lower bound of 12 is suggested in [14] as the minimum number of points for analysis. An upper bound of 36 (the number of years for which data is available) is used as the upper bound for number of trigonometric terms. To ensure the best fit was obtained, a sensitivity analysis of the control parameter settings for DE was done. The cross over constants (Cr) and the mutation scale factor (F) were varied from 0 to 1 in steps of 0.1 and DE was allowed to run for 1000 generations.

2.5 Comparison of Streamflow Models

Performance Metric I

The mathematical expressions obtained for each function were used to estimate streamflows for different return periods and used in performing the statistical goodness-of-fit tests for the selection of the best fit model. In this study, for comparison of nonparametric and parametric methods, the Mean Relative Deviation (MRD) and Mean Square Relative Deviation (MSRD) statistics were used to measure the goodness-of-fit of the developed models. These statistical terms are defined as follows [3, 13, 15]:

$$MRD = \frac{1}{n}\sum_{i=1}^{n}\frac{|\, x_i - \hat{x}_i \,|}{x_i} \times 100 \tag{5}$$

$$MSRD = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \hat{x}_i}{x_i} \times 100\right)^2 \tag{6}$$

where x and $\hat{x}$ are the observed and estimated streamflows respectively and n is the sample size. In general, lower values of MRD and MSRD indicate better fit [15]. MRD and MSRD values were computed for each developed model, and the model with the minimum values of MRD and MSRD was chosen as the best-fit model.

Performance Metric II

The developed Fourier models were also analyzed statistically with other parametric models using the non parametric Wilcoxon's signed ranks test [22, 23]. A multiple problem analysis comparing models over more than one month simultaneously is performed. This is a pair wise test that aims at detecting significant difference between the behaviors of two algorithms. The null hypothesis for Wilcoxon's test is H0: ΘD=0; in the underlying populations represented by the two samples of results, the median of the difference scores equals zero. The alternative hypothesis is H1: ΘD≠0. In this test, the difference between the performance scores of two models on ith out of N functions are ranked according to their absolute values; average ranks are assigned in case of ties. Let R+ be the sum of ranks for the functions on which the second algorithm (Fourier, in our case) outperformed the first and R-the sum of ranks for the opposite. Let T be the smallest of the sums, T = min(R+, R-). If T is less than or equal to the value of the distribution of Wilcoxon for N degrees of freedom, the null hypothesis of equality of means is rejected. The p-value associated to a comparison is performed by means of the normal approximation for the Wilcoxon T statistics. In this study, SPSS software package was used for computing the p-values. The level of significance is taken as 0.05 which implies that if the p-value is greater than 0.05, there is no significant difference between the performances of the models [23, 24].

3 Results

The mathematical expressions obtained for the six probability distribution functions for each month are presented in Table 2. The computed values of MRD and MSRD for the parametric distributions and the Fourier model are presented in Tables 3 and 4 respectively. Table 5 presents the summary of the statistical Wilcoxon signed rank test. Graphical plots for the month of April were also made to provide insight as to how well the models fit the data. Figures 1 to 6 present plots for the probability distribution models. Figure 7 presents a plot of an unoptimized Fourier series approximation using a 13 data point analysis employing 21 cosine terms and 17 sine terms chosen haphazardly, while Figure 8 shows the fit for an optimized Fourier approximation model. The decision date for reservoir operation in South Africa is 1 May when reservoir operating analysis is undertaken to decide how the reservoir should be operated in the coming year, therefore Plots for April, which is the last water month for the year were made[16, 25].

4 Discussion of Results

According to the results of the MRD and MSRD statistical goodness of fit tests in Tables 3 and 4 respectively, the Fourier approximation model performed best, yielding values close to zero in all cases. The Fourier model performed better than all the probability distribution models (MRD=1.68e^{-12}, MSRD=2.85e^{-23}) for the month of January. Log-Pearson III performed better than other probability models for this month (MRD=5.1915, MSRD=102.87). The Log-Pearson III distribution also performed best amongst the probability models in fitting to the streamflow of February (MRD=8.4212, MSRD=143.60) while the performance of the Fourier model exceeded its performance (MRD=1.74e^{-12}, MSRD=4.94e^{-23}). For the month of March, it was found that the Log-Pearson III distribution produced lower values of the test statistics than all other probability distribution functions (MRD = 15.3688, MSRD= 509.57). The Fourier model clearly outperform Log-Pearson III and all other functions for this month (MRD=4.41e^{-12}, MSRD=3.46e^{-22}). In April, the Fourier model produced lower values of the test statistics than all other functions (MRD=1.90e^{-12}, MSRD=6.26e^{-23}). Likewise in the month of May, the Fourier model outperformed all other models (MRD=1.84e^{-12}, MSRD=3.05e^{-23}). In the month of June, Fourier approximation model performed better in fitting to the streamflow than all the other models (MRD=1.05e^{-12}, MSRD=1.37e^{-23}) also in July the Fourier model was found to be the best fit model (MRD=4.35e^{-13}, MSRD=1.22e^{-24}). For the month of August the Fourier model performed better than all other models (MRD=8.02e^{-13}, MSRD=6.25e^{-24}). In September Fourier series outperformed all other functions (MRD=1.63e^{-12}, MSRD=4.73e^{-23}). Results of the test statistics also show that the Fourier model is the best fit model for the month of October (MRD=1.14e^{-12}, MSRD=1.60e^{-23}). The Fourier model also emerged as the best fit model for the streamflow of November and December (MRD=7.51e^{-13}, MSRD=4.86e^{-24}) and (MRD=1.54e^{-12}, MSRD=3.76e^{-23}) respectively.

Table 2. Summary of developed probability distribution model equations for each month

	Probability Distributions					
Month	Normal	Log-Normal	Pearson III	Log-Pearson III	Gumbel	Log-Gumbel
Jan	Qp = 539.29 + 802.57K	Qp = Antilog (2.55 + 0.35K)	Qp = 539.29 + 802.57K'	Qp = Antilog (2.55 + 0.35K'')	Qp = 178.13 + 626.01YT	Qp = Antilog (2.39 + 0.27YT)
Feb	Qp = 689.35 + 660.42K	Qp = Antilog (2.64 + 0.44K)	Qp = 689.35 + 660.42K'	Qp = Antilog (2.64 + 0.44K'')	Qp = 392.16 + 515.13YT	Qp = Antilog (2.45 + 0.34YT)
Mar	Qp = 771.99 + 1054.10K	Qp = Antilog (2.63 + 0.50K)	Qp = 771.99 + 1054.10K'	Qp = Antilog (2.63 + 0.50K'')	Qp = 297.64 + 822.20YT	Qp = Antilog (2.41 + 0.39YT)
Apr	Qp = 488.49 + 444.21K	Qp = Antilog (2.50 + 0.46K)	Qp = 488.49 + 444.21K'	Qp = Antilog (2.50 + 0.46K'')	Qp = 288.59 + 346.48YT	Qp = Antilog (2.29 + 0.36YT)
May	Qp = 359.00 + 302.45K	Qp = Antilog (2.43 + 0.34K)	Qp = 359.00 + 302.45K'	Qp = Antilog (2.43 + 0.34K'')	Qp = 222.90 + 235.91YT	Qp = Antilog (2.27 + 0.27YT)
Jun	Qp = 294.29 + 228.02K	Qp = Antilog (2.36 + 0.34K)	Qp = 294.29 + 228.02K'	Qp = Antilog (2.36 + 0.34K'')	Qp = 191.68 + 177.85YT	Qp = Antilog (2.21 + 0.27YT)
Jul	Qp = 298.87 + 213.33K	Qp = Antilog (2.37 + 0.32K)	Qp = 298.87 + 213.33K'	Qp = Antilog (2.37 + 0.32K'')	Qp = 202.87 + 166.40YT	Qp = Antilog (2.23 + 0.25YT)
Aug	Qp = 319.46 + 203.46K	Qp = Antilog (2.42 + 0.30K)	Qp = 319.46 + 203.46K'	Qp = Antilog (2.42 + 0.30K'')	Qp = 227.91 + 158.70YT	Qp = Antilog (2.28 + 0.23YT)
Sep	Qp = 366.47 + 302.30K	Qp = Antilog (2.46 + 0.32K)	Qp = 366.47 + 302.30K'	Qp = Antilog (2.46 + 0.32K'')	Qp = 230.44 + 235.80YT	Qp = Antilog (2.31 + 0.25YT)
Oct	Qp = 385.10 + 335.40K	Qp = Antilog (2.48 + 0.29K)	Qp = 385.10 + 335.40K'	Qp = Antilog (2.48 + 0.29K'')	Qp = 234.17 + 261.61YT	Qp = Antilog (2.35 + 0.23YT)
Nov	Qp = 427.62 + 404.71K	Qp = Antilog (2.48 + 0.35K)	Qp = 427.62 + 404.71K'	Qp = Antilog (2.48 + 0.35K'')	Qp = 245.50 + 315.67YT	Qp = Antilog (2.33 + 0.27YT)
Dec	Qp = 469.37 + 417.93K	Qp = Antilog (2.55 + 0.34K)	Qp = 469.37 + 417.93K'	Qp = Antilog (2.55 + 0.34K'')	Qp = 281.30 + 325.99YT	Qp = Antilog (2.39 + 0.27YT)

Q_p – Estimated streamflow using a probability distribution model

K – Normal variate obtained from Normal distribution tables for a given return period

K' - Variate obtained from Pearson distribution tables for a given return period and skew coefficient

K'' - Variate obtained from Pearson distribution tables for a given return period and the log transform of the skew coefficient

Y_T – Reduced variate computed as $Y_T = -\ln[-\ln(1 - {}^{1}/_{T})]$ for a given return period T.

Table 3. Values of MRD using parametric methods and Fourier series

Month	Normal	Log-Normal	Pearson III	Log-Pearson III	Gumbel	Log-Gumbel	Fourier
Jan	151.2811	12.0405	29.8438	5.1915	115.9675	5.5415	$1.68e^{-12}$
Feb	106.7569	9.0357	33.6418	8.4212	57.9728	21.7487	$1.74e^{-12}$
Mar	280.7478	16.8143	46.8523	15.3688	189.0006	31.6935	$4.41e^{-12}$
Apr	111.2141	15.1363	37.2594	11.8002	53.8665	32.6431	$1.90e^{-12}$
May	62.1407	12.7495	14.1287	12.3576	33.8217	18.2945	$1.84e^{-12}$
Jun	58.6075	23.4848	42.2687	19.2187	30.0653	29.6068	$1.05e^{-12}$
Jul	32.7254	6.9456	6.0170	6.3130	12.7471	16.7173	$4.35e^{-13}$
Aug	27.2471	12.0793	10.8115	11.7120	9.5029	19.5252	$8.02e^{-13}$
Sep	60.1752	12.3599	28.6672	11.3696	32.4488	18.8099	$1.63e^{-12}$
Oct	61.6740	13.1636	20.6614	13.1141	39.4811	14.3298	$1.14e^{-12}$
Nov	74.1772	13.3664	22.7491	11.4624	46.6028	11.9927	$7.51e^{-13}$
Dec	64.0198	10.5458	31.3106	9.3157	35.3434	19.8038	$1.54e^{-12}$

Table 4. Values of MSRD using parametric methods and Fourier series

Month	Normal	Log-Normal	Pearson III	Log-Pearson III	Gumbel	Log-Gumbel	Fourier
Jan	65571.19	278.00	3716.25	102.87	33908.63	105.63	$2.85e^{-23}$
Feb	77071.79	180.97	4752.68	143.60	19768.57	1157.50	$4.94e^{-23}$
Mar	498232.24	596.20	7459.35	509.67	204059.55	1886.04	$3.46e^{-22}$
Apr	99001.37	547.21	7791.63	276.50	20770.51	2549.67	$6.26e^{-23}$
May	17623.63	286.50	280.86	262.22	3983.78	830.26	$3.05e^{-23}$
Jun	20708.34	1234.91	11974.04	622.97	2719.96	2980.24	$1.37e^{-23}$
Jul	5091.53	128.16	65.58	96.93	622.77	591.82	$1.22e^{-24}$
Aug	2711.02	379.92	182.76	230.29	124.90	1115.67	$6.25e^{-24}$
Sep	25110.01	810.86	7313.66	509.38	4710.66	2155.83	$4.73e^{-23}$
Oct	14150.77	315.10	1534.72	328.07	4279.72	546.36	$1.60e^{-23}$
Nov	18430.88	252.21	778.09	193.03	6188.91	265.65	$4.86e^{-24}$
Dec	24984.75	318.16	4208.28	244.84	6297.22	1052.59	$3.76e^{-23}$

Table 5. Results of Wilcoxon signed rank test

Fourier vs.	R^+	R^-	Z	p-value
Normal	300	0	-4.286	0.000
Log-Normal	300	0	-4.286	0.000
Pearson III	300	0	-4.286	0.000
Log-Pearson III	300	0	-4.286	0.000
Gumbel	300	0	-4.286	0.000
Log-Gumbel	300	0	-4.286	0.000

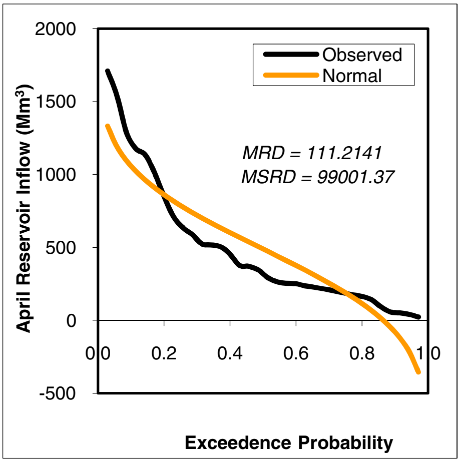


Fig. 1. Observed data and fitted Normal model

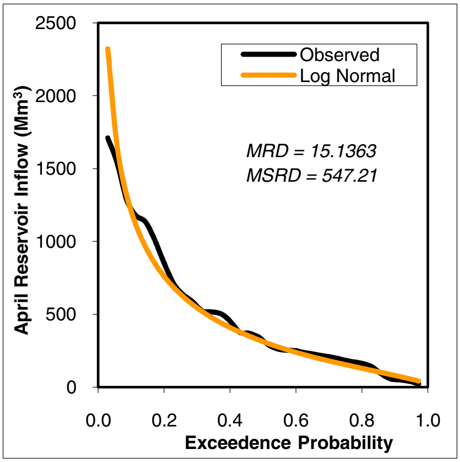


Fig. 2. Observed data and fitted Log-Normal model

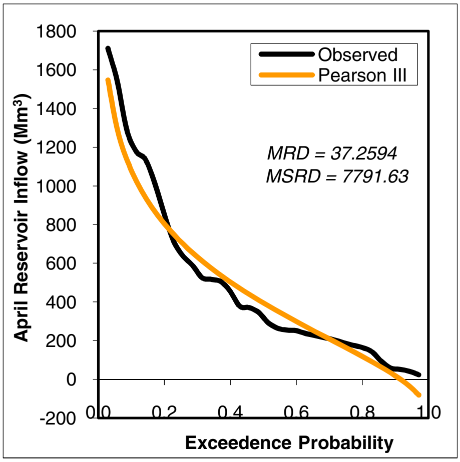


Fig. 3. Observed data and fitted Pearson III model

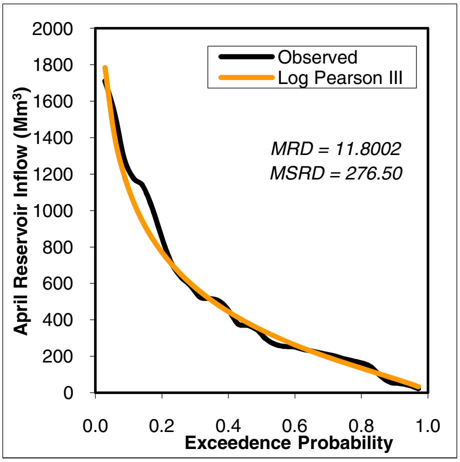


Fig. 4. Observed data and fitted Log-Pearson III model

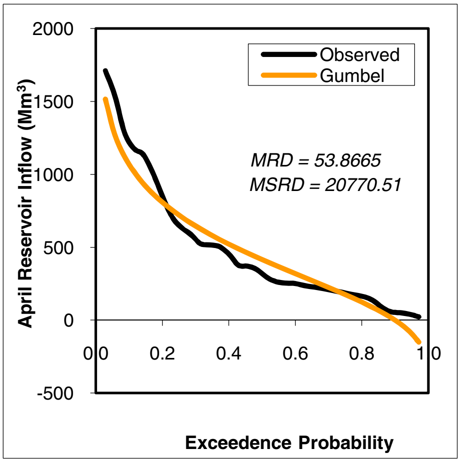


Fig. 5. Observed data and fitted Gumbel model

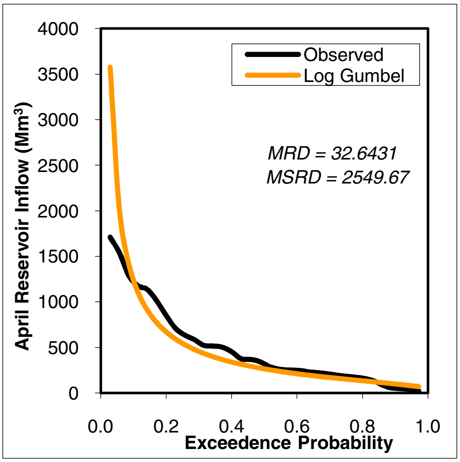


Fig. 6. Observed data and fitted Log-Gumbel model

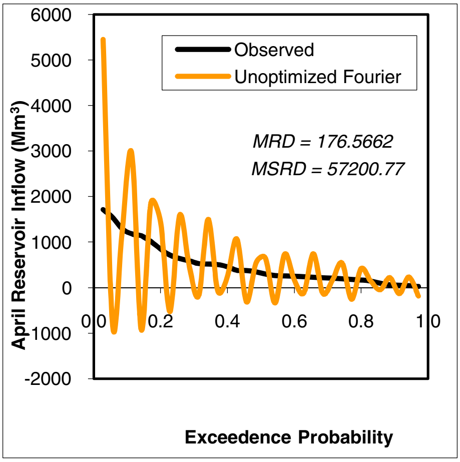


Fig. 7. Observed data and unoptimized Fourier model

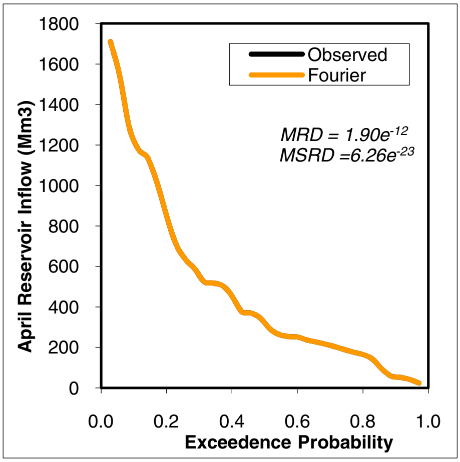


Fig. 8. Observed data and optimized Fourier model

From the results of the Wilcoxon signed rank test in Table 5, it is evident that the difference in the performance of the Fourier approximation models against all the probability distribution models is significant at the 95% and 99% level of significance. Results show that the Fourier model clearly outperformed the Normal model (Z=-4.286, p=0.000), the Log-Normal model (Z=-4.286, p=0.000), Pearson III (Z=-4.286, p=0.000), Log-Pearson III (Z=-4.286, p=0.000), Gumbel (Z=-4.286, p=0.000) and the Log-Gumbel model (Z=-4.286, p=0.000).

In this study, the Fourier approximation function was chosen as the best-fit model as it significantly outperformed all the other functions based on the Wilcoxon test. The Fourier Model performed better in 12 out of the 12 test cases of MRD and

MSRD. Thus the Fourier model performed better in a total of 24 test cases out of 24. Furthermore, plots of the models (see Figures 1 - 6 and 8) show that the Fourier model was able to model the edges of the distribution unlike Log-Pearson III and the other probability distribution models which deviates from the edges of the distribution. Moreover, the number of trigonometrically terms employed for the development of the Fourier models in this study was limited to a maximum of 36, increasing this number may show greater improvements in the performance of the model. This suggests that the Fourier approximation model is adoptable for modeling hydrological processes. This is consistent with the findings of Behnia and Jou, [26] who applied Fourier series to estimate annual flood probability of the Great Karoun river flowing southwest of Iran. The predicted results from the application of the method were compared to results of seven parametric methods including normal, two and three parameter log- normal, two parameter gamma, Pearson and log-Pearson type 3 and Gumbel extreme value type 1. Results of the comparison showed a better ability for Fourier series method. Furthermore, results obtained herein corroborates the findings of Karmakar and Simonovic, [27] who used a 70 years flood peak flow, volume and duration data set for Red River in US and selected the subset of the Fourier series consisting of cosine functions as orthonormal series. They found that the Fourier approximation method is more appropriate than other estimation functions for determining marginal distributions of flood characteristics as it can estimate the probability distribution function over the whole range of possible values. The findings of Jou et al., [15] and Olofintoye and Adeyemo, [13] that the Fourier model performed better in estimating annual rainfall events further support the motion that the Fourier series is adoptable for modeling hydrological processes of streamflow and rainfall.

Inspection of Figure 7 however shows that the Fourier approximation model may fail if the parameters of the model are not properly selected. Selecting the parameters of the model haphazardly may cause the approximation to exhibits typical oscillations that introduce artifacts which are theoretically impossible. Hence, choosing the number of sine and cosine terms haphazardly if therefore not recommended. Selecting the parameters using an optimization procedure like differential evolution is therefore recommended.

5 Conclusion

Detailed analysis of streamflow distribution is paramount in the design, management and operation of hydraulic structures like dams, artificial waterways, drainage networks, retention ponds, and other water-related structures. In this study, six probability distribution models were developed to estimate monthly streamflow in the Vanderkloof dam, South Africa. Fourier approximation models were also fitted to model the distribution of the monthly streamflow series. The models were optimized using the differential evolution algorithm. It was found that the Fourier approximation model was the best-fit model based on Mean Relative Deviation, Mean Squared Relative Deviation goodness-of-fit tests and Wilcoxon signed rank test. It is suggested here that the developed Fourier approximation functions be used for the estimation of monthly streamflow in the Vanderkloof dam. The findings of this study and evidences from previous studies indicate that the Fourier approximation model is adoptable for hydrologic frequency analysis. Therefore, it is concluded that the models developed in this study may be adopted for estimating streamflow in the Vanderkloof dam, South Africa.

References

1. Viessman, W., Lewis, G.L., Knapp, J.W.: Introduction to Hydrology, 3rd edn. Harper and Row Publishers Inc., New York (1989)
2. Jain, A., Srinivasulu, S.: Integrated Approach to Model Decomposed Flow Hydrograph using Artificial Neural Network and Conceptual Technique. Journal of Hydrology 317, 291–306 (2006)
3. 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)
4. Ogunlela, A.O., Oyewole, S.N.: Spectral Analysis of Ilorin Rainfall Data. In: 2nd Annual Civil Engineering Conference, July 26-28. University of Ilorin, Nigeria (2010)
5. Anifowose, M.A., Salami, A.W.: Probability Distribution Models for the Prediction of Meteorological Variables in Ibadan. Journal of Research Information in Civil Engineering 5(1), 30–43 (2008)
6. Filho, A., Santos, C.: Modelling a Dense Urbanized Watershed with an Artificial Neural Network, Weather Radar and Telemetric Data. Journal of Hydrology 317, 31–48 (2006)
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. Mahdavi, M., Khaled, O., Sayed, A.N., Karimi, B., Jalil, M.: Determining Suitable Probability Distribution Models for Annual Precipitation Data (A Case Study of Mazandaran and Golestan Provinces). Journal of Sustainable Development 3(1), 159–168 (2010)
9. Ogunlela, A.O.: Stochastic Analysis of Rainfall Events in Ilorin, Nigeria. Journal of Agricultural Research and Development 1, 39–50 (2001)
10. Wilson, E.M.: Engineering Hydrology, 2nd edn. Macmillan Press Ltd., Houndmills (1969)
11. Zaw, W.T., Naing, T.T.: Modeling of Rainfall Prediction over Myanmar Using Polynomial Regression. In: International Conference on Computer Engineering and Technology (2009)
12. Guo, S.L.: Nonparametric variable kernel estimation with historical floods and paleoflood information. Water Resources Research 27, 91–98 (1991)
13. Olofintoye, O., Adeyemo, J.: Development and Assessment of a Fourier Approximation Model for the Prediction of annual Rainfall in Ilorin, Nigeria. In: Water Institute of Southern Africa Biennial Conference and Exhibition Cape Town, South Africa, May 05-09 (2012)
14. Stroud, K.A.: Further Engineering Mathematics, 3rd edn. Macmillan Press Ltd., London (1996)
15. Jou, P.H., Akhoond-Ali, A.M., Behnia, A., Chinipardaz, R.: A Comparison of Parametric and Nonparametric Density Functions for Estimating Annual Precipitation in Iran. Research Journal of Environmental Sciences 3(1), 62–70 (2009)

16. DWA: Department of Water Affiars. Department of Water Affiars, South Africa (2013), `http://www.dwa.gov.za`
17. Mustapha, S., Yusuf, M.I.: A Textbook of Hydrology and Water Resources, 1st edn. Jenas Prints & Publishing Co., Abuja (1999)
18. Warren, V., Terence, E.H., John, W.K.: Introduction to hydrology, 2nd edn. Intext Educational Publishers, New York (1972)
19. Gerald, C.F., Wheatley, P.O.: Applied Numerical Analysis, 6th edn. Dorling Kindersley (India) Pvt. Ltd., India (2006)
20. Storn, R., Price, K.: Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA (1995)
21. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution APractical Approach to Global Optimization, 1st edn. Springer, Heidelberg (2005)
22. Ali, M., Siarry, P., Pant, M.: An efficient Differential Evolution based algorithm for solving multi-objective optimization problems. European Journal of Operational Research 217(2012), 404–416 (2012)
23. Garcia, S., Molina, D., Lozano, M., Herrera, F.: Study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. Journal of Heuristics 15, 617–644 (2009)
24. Lowry, R.: The Wilcoxon Signed-Rank Test. Concepts & Applications of Inferential Statistics (2013)
25. Mugumo, M.: A Simple Operating Model of the Vanderkloof Reservoir using ANN Streamflow Forecasts. University of the Witwatersrand (2011)
26. Behnia, A.K., Jou, P.H.: Estimating Karoun Annual flood probabilities using Fourier series method. In: 7th International River Engineering Conference. Shahid, Chamran University, Ahwaz, Iran (2007) (in Persian)
27. Karmakar, S., Simonovic, S.P.: Bivariate flood frequency analysis using copula with parametric and nonparametric marginals. In: 4th International Symposium on Flood Defence: Managing Flood Risk, Reliability and Vulnerability, Toronto, Ontario, Canada (2008)

Reservoir Inflow Forecasting Using Differential Evolution Trained Neural Networks

Oluwaseun Oyebode* and Josiah Adeyemo

Department of Civil Engineering and Surveying
Faculty of Engineering and the Built Environment
Durban University of Technology
P.O. Box 1334, Durban, 4000, South Africa
oluwaseun.oyebode@gmail.com

Abstract. This paper presents a study on the application of evolutionary computation and artificial intelligence techniques to forecast inflows into the Vanderkloof reservoir, South Africa for the purpose of planning and management of available water resources. A differential evolution (DE)-trained neural network (DE-NN) was developed to simulate the interaction between reservoir inflow and its causal variables such as precipitation and evaporation. The performance of the DE-NN was evaluated using two performance metrics namely mean absolute percent error (MAPE) and coefficient of determination (R^2). Results from this study demonstrated that the DE-NN model was able to substantially represent inflow patterns into the Vanderkloof reservoir, thereby indicating the efficacy of the DE algorithm in producing adequate generalization on unseen datasets. These results further showcase differential evolution algorithm as a potent, viable and promising algorithm for training neural network models for use in the field of water resources management.

Keywords: Differential evolution, evolutionary computation, learning algorithm, neural networks, reservoir inflow forecasting.

1 Introduction

The management of water resource systems has always been of crucial importance to water managers and decision makers, most especially in water-stressed countries such as South Africa. Water resources engineers and other related stakeholders have developed various approaches towards managing the relatively little amount of water in these regions in order to meet the increasing profile of water demand. However, explosive increase in population continues to place higher demands on the limited water resources. Various techniques have been employed by water resource experts for the purpose of forecasting hydrological variables. These techniques can be categorized majorly into process-based and data-driven models. Although, process-based models

* Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_21, © Springer International Publishing Switzerland 2014

are notably employed in hydrological studies due to their ability to provide detailed representation of the system being modelled, they are however prone to problems of mis-calibration, over-parameterization and parameter redundancy which affect the reliability of their predictions. Data-driven models have however been seen as a promising alternative to the process-based models due to their aptitude to represent the configuration of a system with little or no a priori knowledge. Hence, their usage in water related applications have become increasingly popular.

Neural networks (NNs) are one of the most extensively used data-driven modelling techniques. NN is also a computational intelligence approach which is inspired by neuroscience. NNs can detect and learn correlated patterns between input datasets and corresponding output values [1]. The ability of NNs to undergo training, results in the acquisition of an adaptive learning feature which it uses in solving problems. Over the last two decades, NNs have found application in various fields of water resources which include function approximation, classification and forecasting studies [2-5].

Evolutionary computation (EC) and global optimization techniques have gained much popularity in hydrological modelling studies due to their ability to produce robust models, and also enhance faster convergence towards global optimum. The ease with which they can be integrated into other modelling techniques also serves as a major reason for their prominence. This study aims to investigate the performance a differential evolution (DE)-trained neural networks (DE-NN) for reservoir inflow prediction in the Vanderkloof reservoir, South Africa.

2 Description of Study Area

The Vanderkloof reservoir is the second largest storage reservoir in South Africa with a capacity of over 3 200 million m^3 1 400 km of the Orange River. The Vanderkloof reservoir currently has the highest dam in South Africa with a wall height of 107 m and a crest length of 765 m. The reservoir is situated near Petrusville in the Northern Cape province of South Africa on latitude 29.99222°S and longitude 24.73167°E, with a relief of about 1200 m above mean sea level (AMSL). Average yearly rainfall is between 400 mm and 200 mm while the mean annual temperature is around 18.9^{o}C. It forms an integral component of the Orange River (together with the Gariep reservoir) and supplies water to the Riet River catchment as well as the various users along the remaining reservoir. It is a composite gravity arch dam containing 1.1 million m^3 of concrete and has a central arch which transitions into a gravity flank on the left bank. Four gates installed in the wall can discharge up to 8 500 m^3/s in total through the flood sluices which are positioned on the left flank of the dam [6]. It has a capacity of 3,187,557,000 cubic metres and a surface area of 133.43 square kilometers when full. The reservoir mainly serves to generate hydroelectricity and provide water for irrigation. It is also used for flood control [7].

3 Methodology

3.1 Neural Networks (NNs)

NNs are computational intelligence (CI) techniques inspired by the neurological processing ability of the human brain. NNs consist of a pool of simple processing units called neurons which communicate by sending signals to each other over a large number of weighted connections [8]. The operating principles of NNs is based on parallel distributed information processing that is capable of storing experiential knowledge gained through the process of learning, and making it available for future use [9]. The processing units function by receiving inputs from external sources or other neurons in the network and computing output signals which is transmitted to other units. These processing units are found in layers commonly categorized as input, hidden and output layers. Nonlinearities inherent in the inputs of the system being modelled are transformed into a linear space by the use of an activation function in the hidden layer of the network. The commonly used activation functions are sigmoidal functions such as the logistic and hyperbolic tangent functions [10]. The major network topologies that characterize the architecture of NNs are the feed-forward neural networks (FFNN) and the re-current neural networks; with multilayer perceptron (MLP), radial basis function (RBF) networks, Kohonen's self-organizing feature maps (SOFM) and Elman-type RNN as the most popular NNs [1, 11].

Numerous specialized learning algorithms have been employed for the purpose of training and subjecting NNs to adaptive learning. However, due to the difficulty associated with NN training, optimization algorithms are usually employed for training NNs. These training algorithms either belong to local or global optimization methods. Local methods are usually gradient based, and can either be first-order or second-order methods. The most frequently used training algorithm in the local methods is the back-propagation (BP) algorithm [12], which is based on the first-order gradient of the slope of the objective function. Other approaches which belong to the second-order class include the Levenberg-Marquardt (LM) algorithm and the conjugate gradient (CG) algorithm. The local methods are generally computationally efficient algorithms, and have been adopted in a wide range of studies [4, 11, 13-17]. Kisi and Cigizoglu [15] and Maier et al. [18] however pointed out that the local methods are susceptible to being trapped in local optima and may also generate negative values, if the error surface is fairly rugged.

On the other hand, global optimization methods such as Genetic Algorithm (GA), Differential Algorithm (DE), Evolutionary Programming (EP), Particle Swarm Optimization (PSO), and the Shuffled-Complex Evolution are becoming more popular in hydrology. These global optimization methods have been used to optimize a predefined set of model parameters [19-26], or used in conjunction with the local optimization methods which include the BP, LM, CG and DE algorithms [27-31]. In comparison to the local methods, the global optimization methods have increased ability to overcome local optima and obtain more steady solutions.

The multilayer FFNN architecture was employed in this study. The FFNN was trained using differential evolution (DE) algorithm. A brief introduction on working principles of DE and the rational for its adoption is presented in the next section.

3.2 Differential Evolution (DE)

Differential evolution (DE) is a relatively new technique in evolutionary computation, which was introduced by Storn and Price [32]. DE is a population-based heuristic algorithm for global optimization. The working operation of the DE algorithm primarily involves four stages namely initialization, mutation, recombination and selection. DE differs from other population-based techniques in that it employs differential mutation [33]. Differential mutation as a simple adaptive scheme is used in DE to scale mutation increments to its correct magnitude. In like manner, DE uses a non-uniform crossover in which the parameter values of an offspring vector are inherited in uneven proportions from the parent vectors [34]. Furthermore, the selection method utilized by DE is the tournament selection method, where the offspring vector competes against one or both of its parents [34].

Assuming a population of solutions of size, NP, within a D-dimensional search space and a defined number of real valued vectors are randomly initialized. Two arrays referred to as the primary and secondary arrays are used to hold the population of NP, D-dimensional, real valued arrays. The primary array holds the current vector population, while the secondary array accumulates vectors that are selected for the next generation. Every pair of vectors (X_a, X_b) defines a vector differential: $X_a - X_b$. Upon the random selection of vectors X_a and X_b, they undergo combination via mutation operation so as to produce new generation of vectors (a weighted differential of the vectors). Crossover operation is thereafter performed on the newly generated vectors. This involves the perturbation of the newly generated vectors with a randomly selected target vector, X_c to produce a trial vector. The scaling of the vectors is performed in DE by using a scaling factor, F, which is a user-defined constant in the range of 0 to 1.2. Similarly, the non-uniform crossover is implemented using a user-defined constant in the range $0 \leq CR \leq 1.0$ [34]. The resulting trial vector is chosen for the next generation if and only if it generates a decrease in the value of the objective function. Consequently, evolution of vectors occurs over successive iterations and generations in order to explore the search space, and to locate the minima of the objective function. The iteration is finally brought to a halt when a predefined termination criterion is met.

DE has been successfully applied for solving complex engineering problems due to its ability to diffuse close to the global optimum solution [35-37]. With particular reference to training of NNs, DE have been found to produce better and quicker convergence using only small number of parameters for its algorithm setup, unlike other training algorithms such as BP, LM, CG and GA which are either susceptible to local optima, or require high computational times and huge number of iterations to obtain satisfactory results [23, 29, 34, 38].

3.3 Data Analysis

Historical observations of monthly inflows into the Vanderkloof reservoir for a 27-year period (1986 – 2012) were provided by the Department of Water Affairs, South Africa. In addition, corresponding values of precipitation and evaporation were also provided to cover the period under study. Figures 1, 2 and 3 present plots of the monthly averages of inflows, precipitation and evaporation respectively. It can be observed that the inflow and precipitation values share the same distribution configurations, with peak values of both variables occurring in April 1988 and January 2011; same also for evaporation. It can be inferred that precipitation and evaporation both contribute to the magnitude of inflows in the reservoir.

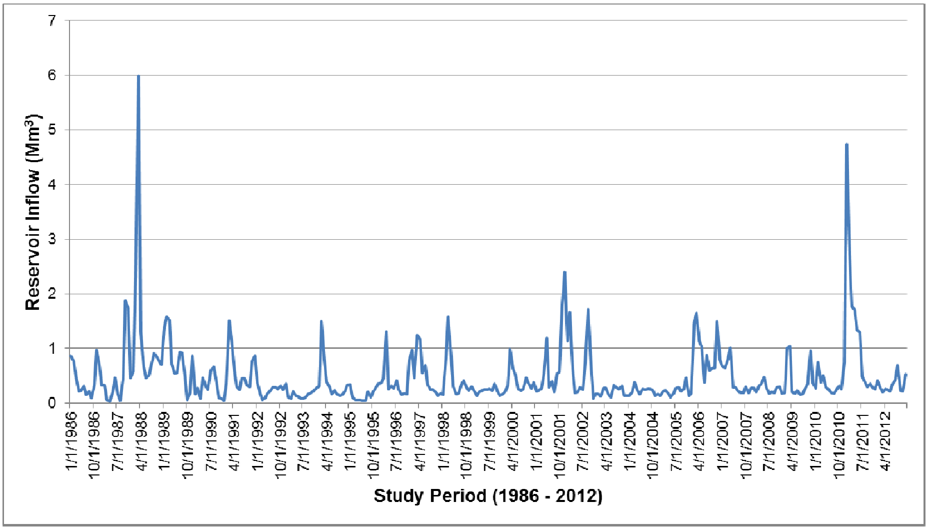


Fig. 1. Plot of monthly values of reservoir inflow over the study period

3.4 Input Variable Selection

To effectively develop a model that would adequately simulate the inflows into the reservoir, it is important to ensure the selection of only input variables that will give the best representation of the system being modelled. It has been established that the inclusion of irrelevant inputs in NN models increases the network size, heightens model complexity, reduces model interpretability, slows down the learning process and consequently leads to mis-convergence [39]. Thus, the determination of the appropriate input variables entails finding the lags of input variables that have a significant influence on the predicted output (inflow). In this study, the dependency between input variables and the associated lag effect were determined using correlation analysis. Correlation analysis has been successfully applied for input variable determination in river hydrology by many researchers [40, 41].

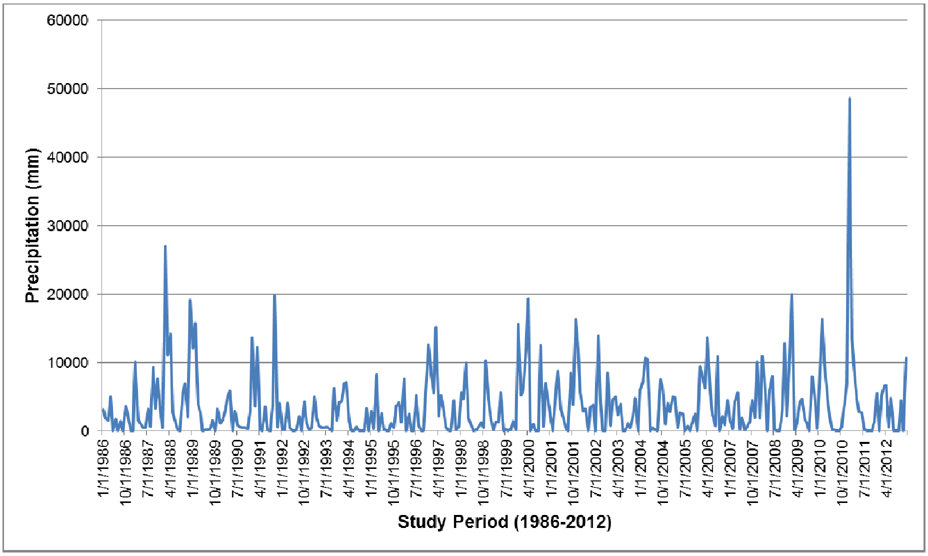


Fig. 2. Plot of monthly values of precipitation over the study period

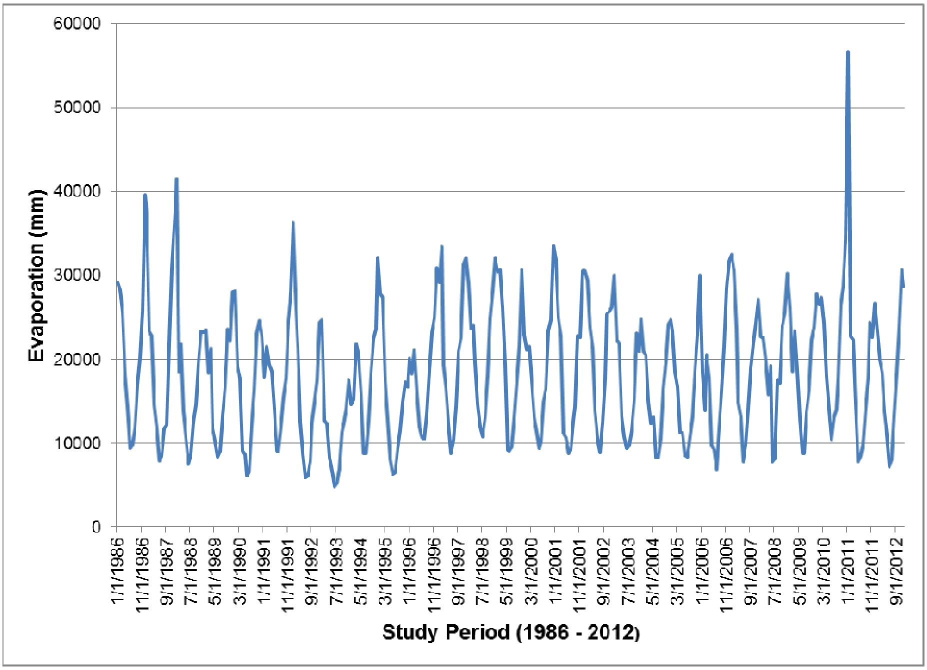


Fig. 3. Plot of monthly values of evaporation over the study period

Results of correlation analysis showed that a relatively high correlation of 0.522 exists between precipitation and reservoir inflow, compared to low correlation values of 0.240 observed between evaporation and reservoir inflows. Thus, the potential input parameters considered for the population of the input vector space were reservoir inflow (Q) and precipitation (P), while evaporation was discarded.

For the purpose of this study, a one (1)-month lead time forecast was adopted for use. The results of correlation analysis between the lags of the input parameters (precipitation and reservoir inflow) and reservoir inflow for the next month (Q_{t+1}) is presented in Table 1 below.

Table 1. Results of correlation analysis showing relationship between potential input parameters and the target output

Input Parameters	Target Output (Q_{t+1})
Q_t	0.55
Q_{t-1}	0.24
Q_{t-2}	0.15
P_t	0.53
P_{t-1}	0.23
P_{t-2}	0.14

where Q_t = streamflow for a given month in a given year; Q_{t-1} = streamflow for the preceding month of a given year.

Q_t and P_t were selected due to higher correlation values relative to other parameters. It was observed that the degree of correlation between the previous inflows and precipitation and the current inflow decreases with increase in the number of lags. However, upon test runs/simulations, Q_t produced lower error estimates when compared to that produced when a combination of Q_t and P_t is used. Therefore, only the inflow for the preceding month Q_t was selected and used to forecast of the inflow for the next month (Q_{t+1}). The mathematical representation of the 1-month lead time model is expressed in equation 1.

$$Q_{t+1} = f(Q_t) \tag{1}$$

Following the construction and formulation of the input vector space of the DE-NN model, the datasets constituted 323 nodes and were thereafter split into two subsets using 70% of the first datasets (226 samples) for model training and the remaining 30% (97 samples) for testing. The training datasets were used for the purpose of model development while the testing datasets were used to check the predictive capability of the models.

3.5 Model Development

As earlier mentioned, the multilayer FFNN was employed for the purpose of forecasting the reservoir inflows in this study. The multilayer FFNN consisted of three layers, with one neuron in each of its input and output layers, representing the selected input variable (Q_t) and the target output (Q_{t+1}) respectively. However, the optimal architecture of the model was determined by varying the number of hidden layer neurons incrementally from 1 to 10 applying a single stepping approach. The DE algorithm, written using Visual Basic for Applications (VBA) programming language was employed for training the network, and was run for 10 000 generations on an Intel Core i3 PC with 2.53GHz and 4GB RAM. The population size, NP, and crossover constant, CR, were used to control the algorithm run, while the mutation scale factor, F, controlled the amplification of differential variation during the run. Following the suggestion of Price and Storn [42], the DE control parameters, NP, CR and F were set at "D multiplied by 10", 0.9 and 0.4 respectively, (where D is the number of weights and biases in the selected architecture).

In terms of data-preprocessing, a logistic sigmoidal-type activation function of between 0 and 1 was used in the hidden layer of the FFNN, to rescale the inputs in the range 0.1 – 0.9. The rescaling of the inputs to the extreme ranges of the activation function was avoided in accordance with the advice of Maier and Dandy [10] who stated that such form of rescaling could reduce the size of the weight updates, thereby resulting into flats spots during training. A linear activation function was however employed in the output layer in order for the network to transform nonlinearities in the inputs into a linear space. In addition, an early-stopping method was introduced to avoid the problem of over-fitting. The early stopping method seeks to identify the point where minimum error on the validation datasets starts to increase, and immediately stops training.

Two (2) standard model evaluation criteria namely, mean absolute percent error (MAPE) and coefficient of determination (R^2) were used to investigate the performance of the model developed in this study. The two model evaluation measures were computed using the mathematical expressions in equations (2) and (3).

1. The Mean Absolute Percent Error (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Q_p - Q_o}{Q_o}\right| \times 100 \tag{2}$$

2. Correlation of Determination (R^2):

$$R^2 = \left[\frac{\Sigma(Q_o - \overline{Q_o})(Q_p - \overline{Q_p})}{\sqrt{\Sigma(Q_o - \overline{Q_o})^2 \Sigma(Q_p - \overline{Q_p})^2}}\right]^2 \tag{3}$$

Q_o and Q_p represent observed and predicted inflows respectively, while $\overline{Q_o}$ and $\overline{Q_p}$ represent their corresponding mean values.

Clearly, lower values of MAPE would indicate better forecast accuracy of the model, while higher values of R^2 (close to 1.0) would indicate better forecast accuracy of the models.

4 Results and Discussions

The results in terms of the aforementioned performance evaluation metrics are presented in Table 2 for both the training and testing datasets. It can be observed that the DE-NN model produced low error estimates in terms MAPE while recording high R^2 values both during the training and testing phases. Furthermore, the results show that the model was able to produce a good generalization on the unseen datasets, as lower error estimate (MAPE = 24.4%) was produced during testing phase as compared to that produced during training (MAPE = 27.8%). In addition, the high R^2 values recorded during the training and testing phases (R^2 = 0.95 & 0.97 respectively) indicates a strong positive correlation between observed and predicted reservoir inflow values. This further proves the ability and consistency of DE algorithm to generalize across the training and unseen datasets as also observed in a water demand forecasting study by Qu et al. [43], where DE was able to produce an adequate representation between water demand and its causal factors.

Table 2. Performance evaluation of the developed model

Performance Metrics	Training Phase	Testing Phase
MAPE (%)	27.8	24.4
R^2	0.95	0.97

It can also be inferred from results produced in this study that the application of early stopping method successfully prevented the occurrence over-fitting and memorization which are problems that typically plague NN models. The early stopping method was considered to have improved the forecast accuracy of the DE-NN model. Furthermore, DE algorithm was able to determine the optimal network architecture of the model, as the architecture with minimum error estimates was returned at the end of the run, thereby ensuring high predictive ability with minimum model complexity. The optimal network architecture of the NN model as determined by the DE algorithm in this study is given as 1-5-1. This implies that the number of hidden layer neurons that gave the best representation of the reservoir inflow pattern is five (5).

Figures 4 and 5 presents plots of observed against predicted reservoir inflows. It can be observed that the model showcased high level of accuracy in simulating the inflow dynamics of the reservoir, most especially for the low and intermediate flows. However, the model was found to under-estimate a few number of peak inflows. This can be considered to be a result of existence of only few extreme events in the pattern of the inflow datasets employed in this study.

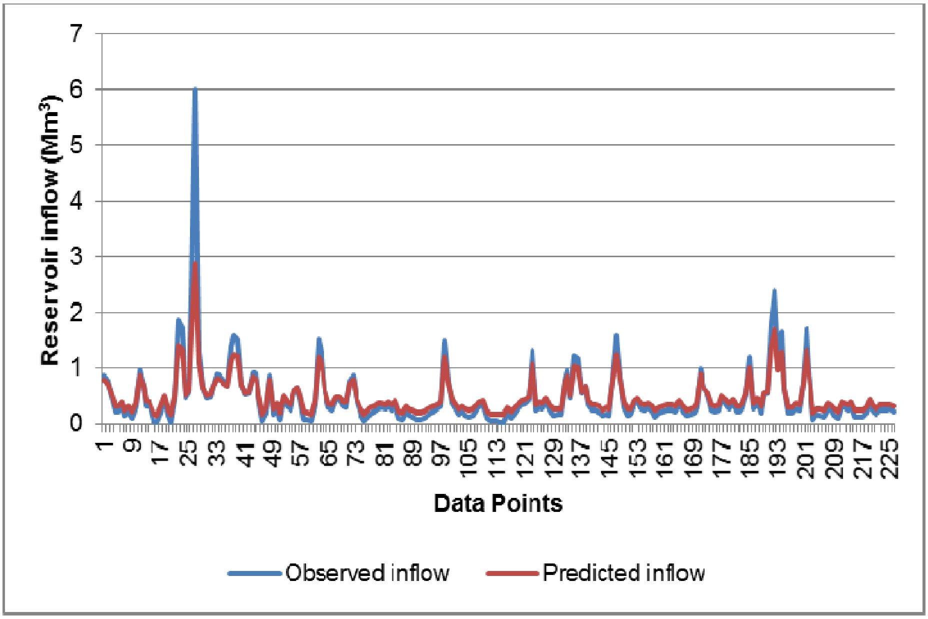


Fig. 4. Plot of observed vs. predicted inflows during training phase

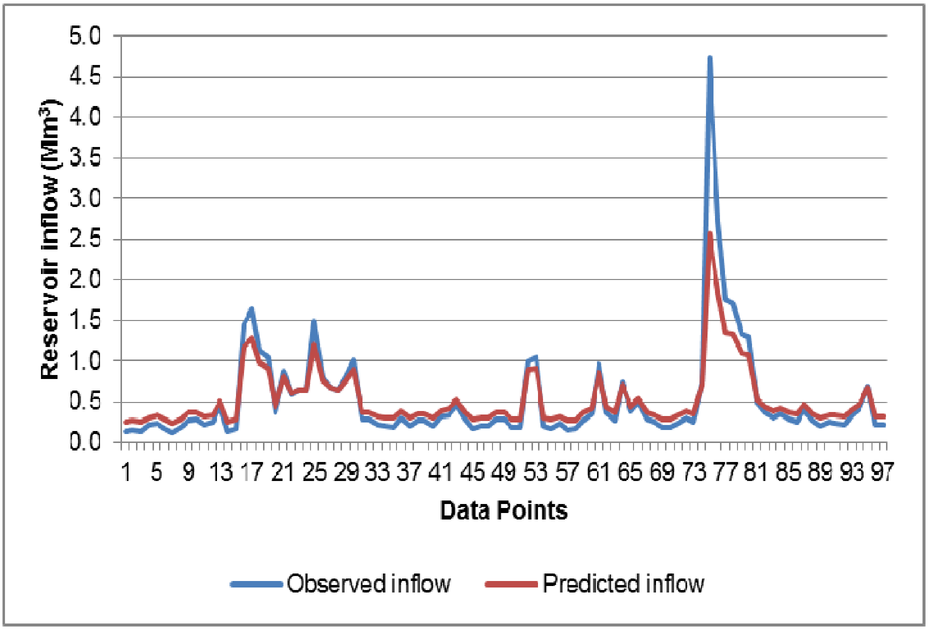


Fig. 5. Plot of observed vs. predicted inflows during testing phase

In all, it can be said that the model was able to adequately reproduce the inflow pattern of the Vanderkloof reservoir with substantial degree of accuracy. This points to the fact that the input variable used in this study have been carefully selected, and the optimization techniques appropriately executed. Results from this study further showcase DE has a promising algorithm that could be used for training of NNs, and most especially in water-related applications.

5 Conclusions

This study presents findings of an investigation of the efficacy of DE in training NNs for the purpose of reservoir inflow forecasting. Results show that the DE-trained NN model developed in this study was able to give a good representation of the inflow dynamics of the Vandekloof reservoir in South Africa. The results further indicate that DE algorithm was efficient in the determination of the optimal network architecture of the NN model – producing minimal errors at least possible model complexity, thereby resulting into high forecast accuracy and substantial reproduction of the inflow pattern of the reservoir. The developed model exhibited good generalization over the training and testing forecast phases. The adoption of the early stopping approach was also found to have contributed to the brilliant performance of the developed model as it ensured avoidance of over-fitting problems. Results from this study further indicate that evolutionary computation techniques such as DE can be easily integrated in any other modelling technique for the purpose of complementary modelling. Future research will focus on fine-tuning the DE algorithm via sensitivity analysis (varying CR and F) to produce better generalization especially as it concerns forecast of extreme events.

References

1. Jha, G.K.: Artificial neural networks and its applications, http://www.iasri.res.in/ebook/ebadat/5-Modeling%20and%20Forecasting%20Techniques%20in%20Agriculture/5-ANN_GKJHA_2007.pdf
2. Coulibaly, P., Anctil, F., Bobee, B.: Multivariate reservoir inflow forecasting using temporal neural networks. Journal of Hydrologic Engineering 6(5), 367–376 (2001)
3. Cigizoglu, H.K.: Application of generalized regression neural networks to intermittent flow forecasting and estimation. Journal of Hydrologic Engineering 10(4), 336–341 (2005)
4. Toth, E.: Classification of hydro-meteorological conditions and multiple artificial neural networks for streamflow forecasting. Hydrology and Earth System Sciences 13(9), 1555–1566 (2009)
5. Yonaba, H., Anctil, F., Fortin, V.: Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. Journal of Hydrologic Engineering 15(4), 275–283 (2010)
6. Adeyemo, J., Otieno, F.O.: Maximization of hydropower using strategies of differential evolution. OIDA International Journal of Sustainable Development 1(2), 33–37 (2010)

7. Department of Water Affairs (DWA) South Africa. Vanderkloof Dam, http://www.dwaf.gov.za/orange/mid_orange/vanderkl.aspx
8. Kröse, B., van der Smagt, P.: An introduction to neural networks, 8th edn. The University of Amsterdam, Amsterdam (1996)
9. Elshorbagy, A., Corzo, G., Srinivasulu, S., Solomatine, D.: Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology—part 1: concepts and methodology. Hydrology and Earth System Sciences 14(10), 1931–1941 (2010)
10. Maier, H.R., Dandy, G.C.: Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental Modelling & Software 15(1), 101–124 (2000)
11. Coulibaly, P., Evora, N.: Comparison of neural network methods for infilling missing daily weather records. Journal of Hydrology 341(1), 27–41 (2007)
12. Rumelhart, D., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: Explorations in the Microstructures of Cognition, pp. 318–362. The MIT Press, Cambridge (1986)
13. Kisi, Ö.: Daily river flow forecasting using artificial neural networks and auto-regressive models. Turkish J. Eng. Env. Sci. 29, 9–20 (2005)
14. Corzo, G., Solomatine, D.: Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. Hydrological Sciences Journal 52(3), 491–507 (2007)
15. Kisi, O., Cigizoglu, H.K.: Comparison of different ANN techniques in river flow prediction. Civil Engineering and Environmental Systems 24(3), 211–231 (2007)
16. Wang, W.-C., Chau, K.-W., Cheng, C.-T., Qiu, L.: A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. Journal of Hydrology 374(3), 294–306 (2009)
17. Ghanbarpour, M.R., Amiri, M., Zarei, M., Darvari, Z.: Comparison of stream flow predicted in a forest watershed using different modelling procedures: ARMA, ANN, SWRRB, and IHACRES models. International Journal of River Basin Management 10(3), 281–292 (2012)
18. Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P.: Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. Environmental Modelling & Software 25(8), 891–909 (2010)
19. Hsu, K.L., Gupta, H.V., Gao, X., Sorooshian, S., Imam, B.: Self ‐ organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. Water Resources Research 38(12), 38-1-38-17 (2002)
20. Dorado, J., RabuñAL, J.R., Pazos, A., Rivero, D., Santos, A., Puertas, J.: Prediction and modeling of the rainfall-runoff transformation of a typical urban basin using ANN and GP. Applied Artificial Intelligence 17(4), 329–343 (2003)
21. Parasuraman, K., Elshorbagy, A.: Cluster-based hydrologic prediction using genetic algorithm-trained neural networks. Journal of Hydrologic Engineering 12(1), 52–62 (2007)
22. Kagoda, P.A., Ndiritu, J., Ntuli, C., Mwaka, B.: Application of radial basis function neural networks to short-term streamflow forecasting. Physics and Chemistry of the Earth, Parts A/B/C 35(13), 571–581 (2010)
23. Piotrowski, A.P., Napiorkowski, J.J.: Optimizing neural networks for river flow forecasting–Evolutionary Computation methods versus the Levenberg–Marquardt approach. Journal of Hydrology 407(1), 12–27 (2011)
24. Dhamge, N.R., Atmapoojya, S., Kadu, M.S.: Genetic Algorithm Driven ANN Model for Runoff Estimation. Procedia Technology 6, 501–508 (2012)

25. Mathur, S.: Particle swarm optimization trained neural network for aquifer parameter estimation. KSCE Journal of Civil Engineering 16(3), 298–307 (2012)
26. Olofintoye, O., Adeyemo, J., Otieno, F.: A Combined Pareto Differential Evolution Approach for Multi-objective Optimization. In: Schuetze, O., Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Moral, P.D., Legrand, P. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III. SCI, vol. 500, pp. 213–231. Springer, Heidelberg (2014)
27. Muleta, M.K., Nicklow, J.W.: Joint application of artificial neural networks and evolutionary algorithms to watershed management. Water Resources Management 18(5), 459–482 (2004)
28. Chen, Y.-H., Chang, F.-J.: Evolutionary artificial neural networks for hydrological systems forecasting. Journal of Hydrology 367(1), 125–137 (2009)
29. Mihalache, C.R., Leon, F.: Functional approximation using neuro-genetic hybrid systems. Buletinul Institutului Politehnic Din Iasi Tomul LV(LIX), 87–102 (2009)
30. Subudhi, B., Jena, D.: An improved differential evolution trained neural network scheme for nonlinear system identification. International Journal of Automation and Computing 6(2), 137–144 (2009)
31. Kişi, Ö.: River suspended sediment concentration modeling using a neural differential evolution approach. Journal of Hydrology 389(1), 227–235 (2010)
32. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
33. Chauhan, N., Ravi, V., Karthik Chandra, D.: Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. Expert Systems with Applications 36(4), 7659–7665 (2009)
34. Abdul-Kader, H.: Neural Networks Training Based on Differential Evolution Algorithm Compared with Other Architectures for Weather Forecasting34. IJCSNS 9(3), 92–99 (2009)
35. Qian, B., Wang, L., Hu, R., Huang, D., Wang, X.: A DE-based approach to no-wait flowshop scheduling. Computers & Industrial Engineering 57(3), 787–805 (2009)
36. Adeyemo, J., Otieno, F.: Differential evolution algorithm for solving multi-objective crop planning model. Agricultural Water Management 97(6), 848–856 (2010)
37. Pal, S., Qu, B., Das, S., Suganthan, P.: Optimal synthesis of linear antenna arrays with multi-objective differential evolution. Progress in Electromagnetics Research, PIER B 21, 87–111 (2010)
38. Slowik, A., Bialko, M.: Training of artificial neural networks using differential evolution algorithm. In: 2008 Conference on Human System Interactions, pp. 60–65. IEEE Xplore (2008)
39. Bowden, G.J., Dandy, G.C., Maier, H.R.: Input determination for neural network models in water resources applications. Part 1—background and methodology. Journal of Hydrology 301(1), 75–92 (2005)
40. Sudheer, K., Gosain, A., Ramasastri, K.: A data‐driven algorithm for constructing artificial neural network rainfall‐runoff models. Hydrological Processes 16(6), 1325–1330 (2002)
41. Aqil, M., Kita, I., Yano, A., Nishiyama, S.: A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. Journal of Hydrology 337(1), 22–34 (2007)
42. Price, K. and Storn, R.: Differential Evolution (DE) for Continuous Function Approximation, `http://www1.icsi.berkeley.edu/~storn/code.html`.
43. Qu, J., Cao, L., Zhou, J.: Differential evolution-optimized general regression neural network and application to forecasting water demand in Yellow River Basin. In: 2010 2nd International Conference on Information Science and Engineering (ICISE), pp. 1129–1132. IEEE Xplore (2010)

Multi-objective Optimization of Methane Producing UASB Reactor Using a Combined Pareto Multi–objective Differential Evolution Algorithm (CPMDE)

Abimbola M. Enitan[1,2,*], Josiah Adeyemo[3], O. Oluwatosin Olofintoye[3],
Faizal Bux[2], and Feroz M. Swalaha[1,2]

[1] Department of Biotechnology and Food Technology,
[2] Institute for Water and Wastewater Technology, Department of Biotechnology,
[3] Department of Civil Engineering and Surveying,
Durban University of Technology, P.O. Box 1334, Durban, 4000, South Africa
enitanabimbola@gmail.com

Abstract. Multi–objective optimization of an operating industrial wastewater treatment plant was carried out using combined Pareto multi–objective differential evolution (CPMDE) algorithm. The algorithm combines methods of Pareto ranking and Pareto dominance selections to implement a novel selection scheme at each generation. Modified methane generation and the Stover–Kincannon kinetic mathematical models were formulated for optimization. The conflicting objective functions that are optimized in this study include, maximization of volumetric methane production rate in the biogas produced at a lower hydraulic retention time and optimum temperature; minimization of effluent substrate concentration in order to meet the environmental discharge requirements based on the standard discharge limit, and finally, the minimization of biomass washout from the reactor. Wastewater flow rate, hydraulic retention time, efficiency of substrate utilization within the reactor, influent substrate concentration and operational temperature are the important decision variables related to this process. A set of non-dominated solutions with the high methane production rate at lower biomass and almost constant solution for the effluent concentration was obtained for the multi-objective optimization problem. In this study, the simulation results showed that the CPMDE approach can generate a better Pareto-front of the selected problem and its ability to solve unconstrained, constrained and real-world optimization problem was also demonstrated.

Keywords: Differential evolution, methane production, multi–objective optimization, Pareto, wastewater treatment plant.

1 Introduction

Environmental pollution, especially water and air pollution, has become a challenging task for both engineers and scientists in the world. The depletion of fossil fuels is

[*] Corresponding author.

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_22, © Springer International Publishing Switzerland 2014

driving intense search into alternative renewable sources, among which is biogas. Biogas is a biofuel (60-70% methane), produced by an anaerobic digestion (AD) of organic waste through synergistic metabolic activities of consortia of hydrolytic, acetogenic and methanogenic bacteria on organic materials [1]. Currently, AD is used to treat more than 10% of organic wastes from the industries for generation of energy in several European countries [1] and saving of chemicals. The industrial viability of this process requires a suitable combination of physical and chemical process parameters and a low-cost substrate; hence there is a need for process optimization.

There are many optimization problems in science and engineering that require maximization of system desirable properties and simultaneously minimizing its undesirable characteristics. A significant portion of research and applications in the field of anaerobic digestion optimization has focused on single-objective optimization problems, whereas most of the natural world problems involve multiple-objectives which are conflicting in nature [2-5]. Multi–objective optimization problem (MOOP) involves finding one or more optimum solutions to more than one objective optimization problem [6]. The aim of MOOPs is to simultaneously optimize a set of conflicting objectives to obtain a group of alternative trade-off solutions called Pareto-optimal or non-inferior solutions which must be considered equivalent in the absence of specialized information concerning the relative importance of the objectives [7, 8].

The optimization problems now-a-days, are represented as an intelligent search problem, where one or more agents are employed to determine the optimal on a search landscape, representing the constrained surface for the optimization problem [9]. A large portion of control problems exhibits multiple stage, multiple objective (MSMO) characteristics. Likewise, AD processes also involve a lot of decision making resulting in many objective functions and constraints. Despite this prevalence, there are few methods with the capability to solve general large-scale conflicting multi-objective optimization problems.

Evolutionary algorithms (EAs) are computational-based biological–inspired optimization algorithms. They are stochastic searching methods, commonly used for solving non-differentiable, non-continuous and multimodal optimization problems based on Darwin's natural selection principle [10, 11]. EAs are widely used for single and multi-objective optimization of anaerobic digestion processes in relation to methane production [2, 3, 12]. EAs use several variables of a problem to provide an optimum solution. EAs can generate Pareto optimal solutions for different anaerobic digestion models with equally good solutions with respect to all objectives; none of the solutions should dominate another [11, 6]. Studies have shown that EAs are good alternative methods for monitoring state variables in biotechnological processes [13, 3, 2].

Some of the most frequently used evolutionary multi-objective optimization algorithms for anaerobic digestion include non–dominated sorting genetic algorithm (NSGA), multi-objective genetic algorithm (MOGA), multi-objective differential evolution algorithm (MDEA), Multiobjective differential evolution (MODE) and multi-objective particle swarm optimization (MOPSO) [8, 12, 2, 14]. Successful applications of DE to batch fermentation process, optimization of non-linear chemical processes, optimization of process synthesis and design problems, optimization of biomass pyrolysis and optimal design of shell and tube heat exchangers have been reported in the literature [15, 2, 16]. Among other improved versions of differential evolution that have been reported in the literature include hybrid differential evolution (HDE) [17], Pareto differential evolution approach (PDEA) [18], MDEA [19], multi-objective differential evolution algorithm (MODEA) [20] and more recently, a combined Pareto multi–objective differential evolution (CPMDE) algorithm [21].

Olofintoye and coworkers [21] developed a combined Pareto multi-objective differential evolution algorithm for solving multi-objective optimization problems. The algorithm combines methods of Pareto ranking and Pareto dominance selections to implement a novel selection scheme at each generation. A combined Pareto multi-objective differential evolution algorithm employee's harmonic average crowding distance measure as against NSGA that implement a crowding distance. The superiority of harmonic average crowding distance was demonstrated by Huang *et. al.* [23] and the ability of the CPMDE algorithm in solving unconstrained, constrained, as well as real-world optimization problems such as engineering problems has been reported in the literature [21, 22]. The strength of CPMDE algorithm in multi-modal function optimization was investigated by Adeyemo and Olofintoye [22]. Their simulation results show that the CPMDE approach can generate a better Pareto-front of the selected problems [21, 22].

In this study, the main aim of optimizing the methane producing UASB reactor using CPMDE algorithm is to determine an improved and provide parameter settings for operating the reactor for more methane generation and better effluent quality. Thus, CPMDE algorithm is employed to optimize methane producing UASB reactor treating brewery wastewater. It is tested to investigate if the algorithm will perform better for real-life optimization problems such as anaerobic treatment of wastewater in a renewable generation system for better and more robust solutions for the decision makers. In recent times, slightly similar problem has been solved by using different industrial wastewater and algorithm, but we have now used an improved algorithm to solve the problem to have better solutions. A modified methane generation model and the Stover-Kincannon kinetic mathematical models [24] were formulated for the optimization. This is the first application of using CPMDE algorithm in the area of anaerobic treatment of brewery wastewater and methane production rate, as well as the first reported multi-objective optimization study of brewery wastewater treatment plant for methane production, effluent substrate and biomass concentration using the CPMDE algorithm.

2 Optimization of UASB Reactor

The optimization problem was formulated for multi-objective optimization problems of an existing plant that have been scaled-down for easy optimization. The optimization problem was formulated for maximization of methane production rate (Y_v), minimization of effluent biomass concentration (X_e) and effluent COD concentration (S_e). The constrained optimization problem is written as;

$$Maximum\, f_1\ (P, \theta_h, T) = \ Y_v \tag{1}$$

$$Minimum\ f_2\ (S_i, Q) = S_e \tag{2}$$

$$Minimize\ f_3\ (S_i, \theta_h,\ Q,\ S_e) =\ X_e \tag{3}$$

Model equations

$$\mathrm{Y_v} = \frac{(1-\mathrm{P})0.516\mathrm{S_i}}{\theta_\mathrm{h}}\left[1 - \frac{0.046}{\frac{[\theta_\mathrm{h}(0.013(\mathrm{T})-0.129)]}{(0.083\theta_\mathrm{h})+1} - 0.956}\right] \tag{4}$$

$$X_e = \frac{0.357(S_i - S_e)}{0.034\theta_h} \tag{5}$$

$$S_e =\ S_i - \frac{18.51S_i}{13.64 + \left({QS_i}/{V_r}\right)} \tag{6}$$

The decision variables are bounded as;

$$S_{i,L}\ \leq\ S_i\ \leq\ S_i^U \tag{7}$$

$$Q_L\ \leq\ Q\ \leq\ Q^U \tag{8}$$

$$\theta_{h,L}\ \leq\ \theta_h\ \leq\ \theta_h^U \tag{9}$$

$$P_L\ \leq\ P\ \leq\ P^U \tag{10}$$

$$T_L\ \leq T \leq\ T^U \tag{11}$$

Subject to constraints

$$X\ \leq\ X_e \tag{12}$$

$$S\ \leq\ S_e \tag{13}$$

$$V\ \leq\ V_r \tag{14}$$

Where, θ_h is the mean hydraulic retention time, S_i and S_e are the influent and effluent COD concentration respectively, while P is the COD removal efficiency. X_e is the concentration of biomass in the discharge effluent (biomass wash-out) and OLR is the organic loading rate. Q represents the influent flow rate of wastewater; T is the operational temperature while V is the desired reactor volume.

In this problem, plant treatment efficiency depends on the sludge concentration in the reactor. Therefore, prevention of sludge or biomass washout from the reactor is needed for effective treatment and to meet the environmental discharge requirements, as well as increasing the methane production rate for biofuel. Important decision variables are shown in Table 2. In equation (5), the desired value for biomass wash out from the reactor was considered as 0.025g/l. The desired reactor volume of the existing plant is 1400 m^3, but for easy optimization and simulation in the pilot-scale reactor, it was scaled–down to 35 m^3. The results of the optimization can then be scaled-up to the actual volume of the large-scale UASB reactor.

The discharge of low effluent COD concentration to meet the standard limits is another important factor for environmental monitoring, as well as produce substantial amount of biogas that is rich in methane composition. Therefore, it is logical to get an optimum operating condition that minimizes the effluent discharge COD for any OLR, Q and S_i. Thus, minimization of effluent substrate concentration using the modified Stover-Kincannon kinetic model (equation (6)) was included in the optimization. In equation (6), the desired effluent COD concentration is considered as 0.05 g/l.

Table 1. Details of the optimization problem studied

Objective function	**Problem**
First	Maximixe Y_v
Second	Minimize S_e
Third	Minimize X_e
Inequality Constraints	
V_r (L)	= 35
S_e (g/L)	≤ 0.05
X_e (g/L)	≤ 0.025
Bounds	
S_i (g/L)	$1 \leq S_i \leq 10$
Q (L/day)	$1 \leq Q \leq 20$
θ_h (h)	$1 \leq \theta_h \leq 12$
P	$0.8 \leq P \leq 1$
T (˚C)	$10 \leq T \leq 35$

The boundary conditions for the decision variables based on the scaled-down industrial process are shown in Table 1. The lower and upper limits on θ_h were decided based on the formation on hydraulic retention time of the industrial treatment plant. The microbial consortiums in the treatment plant are sensitive to temperature changes, which in turn can affect the rate of methane production. Therefore, operating temperature should be considered as one of the important factors. In this regard, the minimum and maximum values of temperature were selected based on the operating range of the industrial plant.

The lower and upper limits for the influent substrate concentration were set based on the capacity of the treatment plant. The minimum and maximum values for the efficiency of substrate utilization of the reactor at the end of the treatment period in terms of COD removal were considered. This was to ensure maximum conversion of organic matter to methane and good effluent quality in order to meet the discharge standard. The lower and upper limits for influent flow rate were chosen based on the industrial activities and the wastewater the industry is producing, however the volume in this study was scaled-down to 35 m^3 for the pilot-scale reactor.

3 Combined Pareto Multi-objective Differential Evolution (CPMDE) Algorithm

3.1 The CPMDE Algorithm

In this study, combined Pareto multi-objective differential evolution (CPMDE) algorithm was used to optimize the formulated mathematical models. The algorithm combines methods of Pareto ranking and Pareto dominance selections to implement a novel selection scheme at each generation [21]. At each iteration of CPMDE, the combined population of trial and target solutions is checked for non-dominated solutions. Solutions that will proceed to the next generation are selected using a combined Pareto ranking and Pareto dominance selection scheme [25]. After generating a trial population, tournaments are played between trial solutions and their counterparts in the target population at the same index. Diversity among solutions in the obtained non-dominated set is promoted using a harmonic average crowding distance measure [23, 21] to select the solution that will proceed to the next generation if solutions are feasible and non-dominated with respect to each other.

In CPMDE, boundary constraints are handled using the bounce-back strategy [26]. This strategy replaces a vector that has exceeded one or more of its bounds by a valid vector that satisfies all boundary constraints. In contrasts to random re-initialization, the bounce-back strategy takes the progress towards the optimum into account by selecting a parameter value that lies between the base vector parameter value and the bound being violated [2]. Equality and inequality constraints are handled using the constrained-domination technique suggested by Deb [27]. DE/rand/1/bin variant of DE is used as the base for CPMDE. The CPMDE algorithm is summarized as follows [21]:

1. Input the required DE parameters like number of individuals in the population (Np), mutation scale factor (F), crossover probability (Cr), maximum number of iterations/generations (gMax), number of objective functions (k), number of decision variables/parameters (D), upper and lower bounds of each variable, etc.
2. Initialize all solution vectors randomly within the limits of the variable bounds.
3. Set the generation counter, g =0
4. Generate a trial population of size Np using DE's mutation and crossover operations [26]

```
5. Perform a domination check on the combined trial and target population and mark all non-dominated solutions as "non-dominated" while marking others as "dominated".
6. Play domination tournament at each population index.
    i. If the trial solution is marked "non-dominated" and the target is marked "dominated" then the trial vector replaces the target vector.
    ii. If the trial solution is marked "dominated" and the target is marked "non-dominated" then the trial vector is discarded.
    iii. If both solutions are marked "dominated", then replace the target vector if it is dominated by the trial vector or if they are non-dominated with respect to each other.
    iv. If both vectors are marked "non-dominated", then note down the index and proceed to the next index. When all solutions marked "non-dominated" from steps i – iii above are installed in the next generation, then sort out all solutions noted in step iv one at a time using the harmonic average crowding distance measure [23]. The solution with a greater harmonic average distance is selected to proceed to the next generation.
7. Increase the generation counter, g, by 1. i.e. $g = g+1$.
8. If $g < gMax$, then go to step 4 above else go to step 9
9. Remove the dominated solutions in the last generation
10. Output the non-dominated solutions.
*Note domination checks are performed using the naive and slow method suggested by [27].
```

Source: [21].

Olofintoye *et. al.* [21] evaluated the performance of CPMDE using common difficult test problems obtained from multi-objective evolutionary computation literatures. The ability of the algorithm in solving unconstrained, constrained and real optimization problems was demonstrated and competitive results obtained from its application suggest that it is a good alternative for solving multi-objective optimization problems. Furthermore, based on an argument by Deb [6] that most of these test problems are not tuneable and it is difficult to establish the feature of an algorithm that has been tested, CPMDE has further been tested using on tuneable multi-objective test problems [22]. CPMDE has been applied to solve real world multi-objective problems and results obtained corroborate the efficacy of CPMDE in solving multi-objective optimization problems.

3.2 Implementation of CPMDE Algorithm for an UASB Reactor

The ability of CPMDE in solving unconstrained, constrained and real-world anaerobic digestion optimization problems is demonstrated herein. The principle of CPMDE algorithm includes coding of the models, decision variables, the constraints as well as evaluation of the fitness function and improvement of the fitness function using differential evolution operators such as tournament selection, crossover and the harmonic average crowding distance measure. The crossover constant, Cr and the mutation scaling factor, F were set at 0.1 and 0.9. Population size, Np was set to 50 and the algorithm was run for a maximum number of generations, gMax from 300-5000 on different optimization problems. Harmonic average crowding distances were computed using two nearest neighbours. Further details on the implementation of CPMDE may be found elsewhere [21, 22].

4 Results and Discussion

Kinetic model for methane production rate of volumetric methane production rate effluent substrate COD and biomass concentration were simultaneously optimized in this study to get global optimal solutions from the conversion of organic matter in the brewery wastewater. The kinetic coefficient of the model equations used is summarized in Table 1. These models were optimized by using combined Pareto multi-objective differential evolution algorithm on a computer with dual core processor and 8 GB RAM processor. The model equations were first coded and tested with MATLAB software to make sure the code is free of error (results not shown here). Subsequently, CPMDE algorithm was used to solve the models as multi-objective optimization problem.

A multi–objective optimization problem involving three-objective functions was solved simultaneously using CPMDE algorithm. These include (i) maximization of volumetric methane production rate, (ii) minimization of effluent discharge COD and (iii) minimization of biomass wash-out from the treatment plant. For this problem, the constraints and decision variables used are shown in Table 1. The best value of CPMDE optimization parameters for the three–objective functions are shown in Table 2. Fig.1 shows the Pareto optimal solutions for these three-objective functions. Equally good solutions with regard to all objectives were obtained for this problem; none of the solutions dominated another. As the volumetric methane production rate increases (improves), both the effluent discharge COD and biomass wash-out from the treatment plant also increases (worsens) over the entire Pareto optimal surface [28, 11, 6, 3]. From these results, none of the solutions dominate another. All the solutions on the Pareto front are equally good and are expected to provide flexibility for the solutions on the Pareto front. Each point on the Pareto optimal front corresponds to a set of decision variables as shown in Table 1. Some of the advantages of using these three-objective optimization problem is to have a wide choice of solutions and operating points in the Pareto set, because each point on the Pareto set is obtained from a set of decision variables.

The decision variables were further plotted against volumetric methane production rate and effluent biomass concentration to determine the conflicting variables (Fig. 2a-d). However, we noticed a nearly constant decision variables (T, S_i and Q) over the range of Pareto set, thus the results are not plotted. In addition, the degree of scatter of θ_h and P are slightly higher with unsmooth Pareto front for the simultaneous optimization of these objective functions. Similar result was reported by Babu et al. [2] when

MODE and NSGA algorithms were employed for solving multi–objective optimization problems of industrial adiabatic styrene reactor. Iqbal and Guria [3] explained that scattered optimal values of the decision variables compensate for each other due to additional objective function and decision variable to the optimization problem. However, CPMDE is able to give a more uniform distribution of solutions, than those reported by Yee *et. al.* [29] and Babu *et. al. [*2] using NSGA and MODE respectively. Furthermore, several other studies have been reported to have encountered scattered decision variables [30-32]. Better spread in this study shows that CPMDE algorithm found more operating policies that were not discovered by any other algorithms from which the decision maker can choose from. That is, we have more options for operating the reactor to produce more methane during anaerobic degradation of industrial wastewater. In addition, the methane production rate is observed to increase due to an increase in hydraulic retention time. This suggests that the higher the time the wastewater spend in the reactor, the higher the gas production in the reactor. The optimal values of methane production rate take the upper bound at different θ_h and high substrate removal efficiency. At higher effluent flow rate (Q = 14 L/day,V_r = 35), optimal θ_h takes almost lower limit between 8-9 h, and increase in Y_v is observed as θ_h decreases.

In Fig. 2(c), it is noted that the X_e decreases with increase in hydraulic retention time as the COD removal efficiency (P) remains high (Fig. 2d). It may be deduced from the optimal results that high P value between 85-87% and 8-9 h hydraulic retention time at 30-31˚C is responsible for the low and almost constant effluent substrate and biomass concentration with high methane production rate. This suggests that at high influent substrate concentrations and flow rate, high COD removal efficiency and Y_v depend on the time the wastewater spends in an anaerobic reactor. The results further show that the decision variables at mesophilic temperature are responsible for the scattered Pareto solutions in the optimized problems for the three–objective functions as shown in Fig. 2(a-c). Hence, the simulation models could be used to check the operational parameters for getting the best effluent quality, biomass washout and the highest methane production rate in the UASB reactor for brewery wastewater treatment.

Consequently, the multi–objective optimization conditions within the framework of objective functions based on the holistic kinetic models using CPMDE algorithm demonstrated a useful instrument for simultaneous optimization of various operational parameters needed for successful running of an UASB reactor. The strength of the integrated multi-objective optimization approach in this study can be employed for large-scale applications (from pilot–to full–scale reactor). It should be noted that the holistic approach presented in this study is restricted by some boundary conditions and assumptions. However, it can readily be used as a preliminary analysis before transferring the initial concepts to the full–scale reactor, since the model coefficient are obtained from the data collected from the full–scale reactor. Based on the present optimization study, a set of optimal operating conditions is obtained which can enhance the plant performance without affecting the plant configuration. With regard to these facts, future works can consider scaling-up the results obtained in this study to full–scale system.

Table 2. CPMDE parameters used for multi-objective optimization problem

Parameters			Value
Number of Vectors:			50
Number of Parameters:			5
Number of DE generations:			5000
DE control parameters:	Cr	F	
Value	0.1- 0.9	0.1- 0.9	
Step	0.1	0.1	
Optimization			
Number of objectives:			3
Number of constraints:			4
Number of nearest neigbours:			2
Number of non-dominated solutions in final current population			50
Computational time, min			3.17

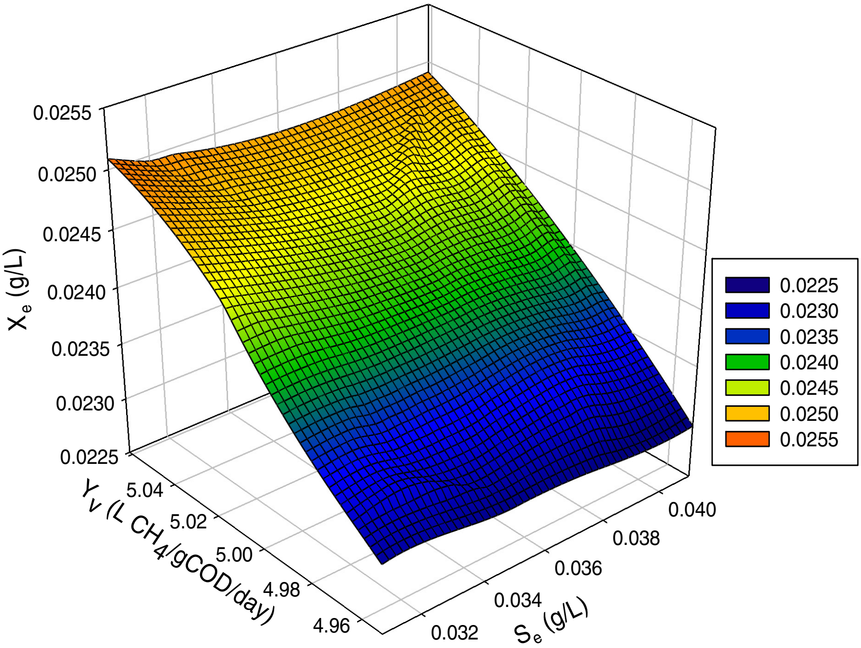


Fig. 1. Pareto optimal set of solutions obtained for the simultaneous optimization of volumetric methane production rate, (Y_v), effluent biomass concentration, (X_e) and effluent substrate concentration, (S_e) as a multi–objective optimization problem

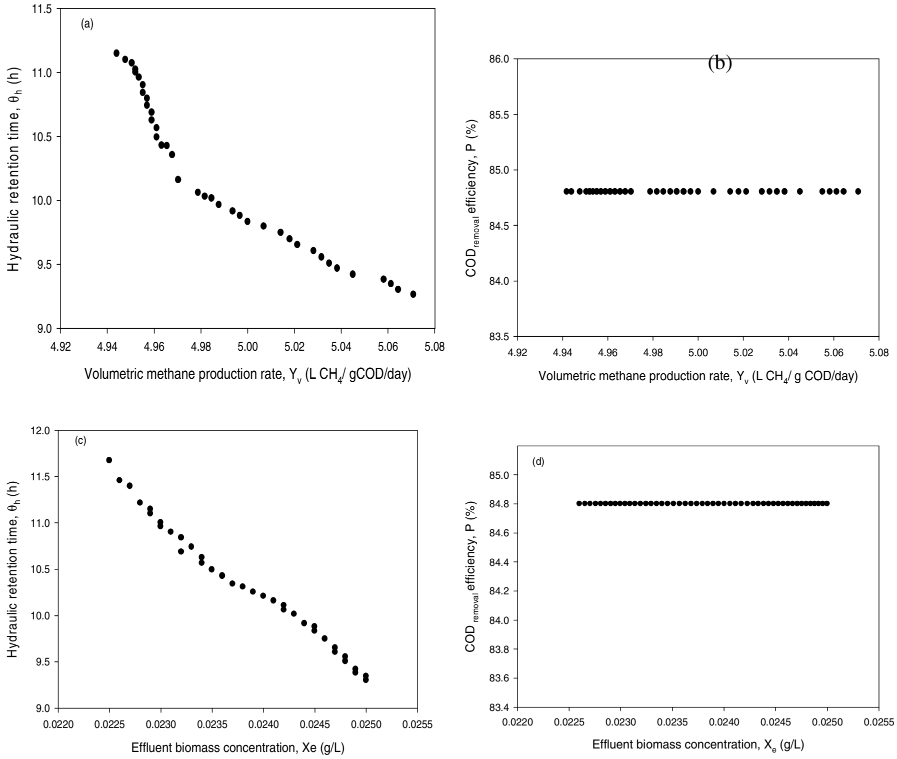


Fig. 2. Optimal variation of the decision variables with Y_v and X_e for the optimized problem

5 Conclusions

It is a known fact that there is a need to develop a novel method for the optimization of complex multi–objective anaerobic digestion process. In this study, optimization of industrial wastewater treatment plant is carried out using combined Pareto multi-objective differential evolution algorithm. Modified methane generation and the Stover–Kincannon kinetic models were formulated for the optimization of anaerobic reactor treating brewery wastewater. Multi–objective optimization problem was solved in this study using CPMDE algorithm as the optimization tool in order to determine the optimal operating conditions of anaerobic reactor treating brewery wastewater. The associated objective functions are: (i) the maximization of volumetric methane production rate, (ii) minimization of discharge effluent substrate concentration, (iii) the minimization of biomass washout from the reactor. Pareto–optimal sets of equally good non-dominated solutions are obtained for the multi–objective optimization problem considered. The decision variables followed the same trend that further proves the reliability of the results obtained in this study. It also shows that the objectives can further be improved. However, it is difficult to compare the results obtained in this study with other studies in the literature due to different substrate and decision variables involved, as well as the algorithm used. Due to the fact that, this study is the first application of using combined Pareto multi–objective differential evolution algorithm for anaerobic digestion optimization of brewery wastewater for better methane production and effluent quality.

The simulation results showed that the CPMDE algorithm can generate a better Pareto-front for the selected problem. Its ability to solve unconstrained, constrained and real-world optimization problem was also demonstrated. This will benefit the existing reactors and the design of new reactors treating brewery wastewater in order use an optimum environmental conditions that will favour the growth of microorganisms in breaking down of complex organic material in the industrial wastewater to specific and desirable end-products, thus enhance biomethane production as source of renewable energy. This study will further help in environmental protection, as well as the generation of energy from anaerobic digestion of industrial wastewater. The outcome could contribute to a sustainable long-term clean development mechanism to generate methane in the biogas produced during the treatment of wastewater in UASB reactors. The captured methane can then be used as fuel and thus reduces methane emissions and obtain a certified emission reduction (CER) credit under the Kyoto Protocol. Hence, the optimization method presented in this study is quite general and flexible to improve the reliability of design and performance of an anaerobic treatment plant either for an existing plant or new ones. It can be applied to any industrial wastewater treatment plant to enhance its robustness and performance in treating high strength wastewater for better discharge effluent quality and biogas production with high methane content for biofuel.

References

1. Gueguim Kana, E.B., Oloke, J.K., Lateef, A., Adesiyan, M.O.: Modeling And Optimization of Biogas Production on Saw Dust and Other Co-Substrates Using Artificial Neural Network and Genetic Algorithm. Renewable Energy (2012), doi:10.1016/j.renene.2012.03.027
2. Babu, B.V., Chakole, P.G., Mubeen, J.H.S.: Multiobjective Differential Evolution (MODE) for Optimization of Adiabatic Styrene Reactor. Chemical Engineering Science 60(17), 4822–4837 (2005)
3. Iqbal, J., Guria, C.: Optimization of an Operating Domestic Wastewater Treatment Plant Using Elitist Non-Dominated Sorting Genetic Algorithm. Chemical Engineering Research and Design 87(11), 1481–1496 (2009)
4. Kusiak, A., Zheng, H.Y., Song, Z.: Wind Farm Power Prediction: A Data-Mining Approach. Wind Energy 12(3), 275–293 (2009)
5. Abu Qdais, H., Bani Hani, K., Shatnawi, N.: Modeling and Optimization of Biogas Production From a Waste Digester Using Artificial Neural Network and Genetic Algorithm. Resources, Conservation and Recycling 54(6), 359–363 (2010), doi:10.1016/j.resconrec.2009.08.012
6. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, 1st edn. John Wiley & Sons Ltd., Chichester (2001)
7. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction. In: Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing, pp. 1–24 (2011)

8. Adeyemo, J., Otieno, F.: Differential Evolution Algorithm for Solving Multi-Objective Crop Planning Model. Agricultural Water Management 97(6), 848–856 (2010), `http://dx.doi.org/10.1016/j.agwat.2010.01.013`
9. Das, S., Abraham, A., Konar, A.: Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives. In: Liu, Y., Sun, A., Loh, H.T., Lu, W.F., Lim, E.-P. (eds.) Advances of Computational Intelligence in Industrial Systems. SCI, vol. 116, pp. 1–38. Springer, Heidelberg (2008), `http://www.springerlink.com`
10. Sendrescu, D.: Parameter Identification of Anaerobic Wastewater Treatment Bioprocesses Using Particle Swarm Optimization. Mathematical Problems in Engineering 2013, 8 (2013), doi:10.1155/2013/103748
11. Enitan, A.M., Adeyemo, J.: Food Processing Optimization Using Evolutionary Algorithms. African Journal of Biotechnology 10(72), 16120–16127 (2011)
12. Wei, X., Kusiak, A.: Optimization of Biogas Production Process in a Wastewater Treatment Plant. In: Proceedings of the 2012 Industrial and Systems Engineering Research Conference (2012)
13. Soons, Z.I.T.A., Streefland, M., van Straten, G., van Boxtel, A.J.B.: Assessment of Near Infrared and “Software Sensor” for Biomass Monitoring and Control. Chemometrics and Intelligent Laboratory Systems 94, 166–174 (2008)
14. Srinivas, N., Deb, K.: Multi-Objective Function Optimization Using NSGA. Evolutionary Computation 2(3), 221–248 (1994)
15. Angira, R., Babu, B.V.: Optimization of Process Synthesis and Design Problems: A Modified Differential Evolution Approach. Chemical Engineering Science 61, 4707–4721 (2006)
16. Babu, B.V., Chaurasia, A.S.: Optimization of Pyrolysis of Biomass Using Differential Evolution Approach. In: Proceeding of Fourth Asia- Pacific Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS), Singapore, December 15-18 (2003)
17. Tsai, K.-Y., Wang, F.-S.: Evolutionary Optimization with Data Collocation for Reverse Engineering of Biological Networks. Bioinformatics 21(7), 1180–1188 (2005), doi:10.1093/bioinformatics
18. Madavan, N.K.: Multiobjective optimization using a Pareto differential evolution approach. Paper presented at the Proceedings of the Congress on Evolutionary Computation, CEC 2002 (2002)
19. Adeyemo, J., Otieno, F.: Multi-Objective Differential Evolution Algorithm for Solving Engineering Problems. Journal of Applied Sciences 9(20), 3652–3661 (2009)
20. Ali, M., Siarry, P., Pant, M.: An Efficient Differential Evolution Based Algorithm for Solving Multi-Objective Optimization Problems. European Journal of Operational Research 217(2), 404–416 (2012), `http://dx.doi.org/10.1016/j.ejor.2011.09.025`
21. Olofintoye, O., Adeyemo, J., Otieno, F.: A Combined Pareto Differential Evolution Approach for Multi-objective Optimization. In: Schuetze, O., Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Moral, P.D., Legrand, P., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III. Studies in Computational Intelligence, vol. 500, pp. 213–231. Springer, Heidelberg (2014)
22. Adeyemo, J.A., Olofintoye, O.O., Otieno, F.A.O.: Performance evaluation of combined pareto multi-objective differential evolution on tuneable multi-objective test beds. International Jounal of Simulation Modeling, DAAAM Int. Vienna (2014)

23. Huang, V.L., Suganthan, P.N., Qin, A.K., Baskar, S.: Multiobjective Differential Evolution with External Archive and Harmonic Distance-Based Diversity Measure. School of Electrical and Electronic Engineering Nanyang, Technological University Technical Report (2005)
24. Enitan, A.M., Kumari, S., Swalaha, F.M., Adeyemo, J., Ramdhani, N., Bux, F.: Kinetic Modelling and Characterization of Microbial Community Present in a Full-Scale UASB Reactor Treating Brewery Effluent. Microbial Ecology 67, 358–368 (2014), doi:10.1007/s00248-013-0333-x
25. Mezura-Montes, E., Reyes-Sierra, M., Coello, C.: Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art. In: Chakraborty, U.K. (ed.) Advances in Differential Evolution. SCI, vol. 143, pp. 173–196. Springer, Heidelberg (2008)
26. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution APractical Approach to Global Optimization, 1st edn. Springer, Heidelberg (2005)
27. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms, 1st edn., vol. 289. John Wiley & Sons, Ltd., Chichester (2001)
28. Liu, P.-K., Wang, F.-S.: Inference of Biochemical Network Models in S-System Using Multiobjective Optimization Approach. Bioinformatics 24(8), 1085–1092 (2008), 1010.1093/bioinformatics/btn1075
29. Yee, A.K.Y., Ray, A.K., Rangiah, G.P.: Multi-objective Optimization of Industrial Styrene Reactor. Computers and Chemical Engineering 27, 111–130 (2003)
30. Tarafder, A., Rangaiah, G.P., Ray, A.K.: Multi-Objective Optimization of an Industrial Styrene Monomer Manufacturing Process. Chemical Engineering Science 60(2), 347–363 (2005)
31. Sareen, R., Gupta, S.K.: Multi-objective Optimization of an Industrial Semi Batch Nylon 6 Reactor. Journal of Applied Polymer Science 58(13), 2357–2371 (1995)
32. Khosla, D.K., Gupta, S.K., Saraf, D.N.: Multi-objective Optimization of Fuel Oil Blending Using the Jumping Gene Adaptation of Genetic Algorithm. Fuel Process Technology 88, 51–63 (2007)

Author Index