Ricardo H.C. Takahashi
Kalyanmoy Deb
Elizabeth F. Wanner
Salvatore Greco (Eds.)

LNCS 6576

Evolutionary Multi-Criterion Optimization

6th International Conference, EMO 2011
Ouro Preto, Brazil, April 2011
Proceedings

Springer

Lecture Notes in Computer Science 6576

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Ricardo H.C. Takahashi Kalyanmoy Deb
Elizabeth F. Wanner Salvatore Greco (Eds.)

Evolutionary Multi-Criterion Optimization

6th International Conference, EMO 2011
Ouro Preto, Brazil, April 5-8, 2011
Proceedings

Springer

Volume Editors

Ricardo H.C. Takahashi
Federal University of Minas Gerais, Department of Mathematics
Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG, Brazil
E-mail: taka@mat.ufmg.br

Kalyanmoy Deb
Indian Institute of Technology Kanpur, Department of Mechanical Engineering
Kanpur 208 016, India
E-mail: deb@iitk.ac.in

Elizabeth F. Wanner
Centro Federal de Educação Tecnológica de Minas Gerais
Department of Computer Engineering
Av. Amazonas 5253, 30421-169 Belo Horizonte, MG, Brazil
E-mail: efwanner@dppg.cefetmg.br

Salvatore Greco
University of Catania, Faculty of Economics
Corso Italia 55, 95129 Catania, Italy
E-mail: salgreco@unict.it

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-19892-2 e-ISBN 978-3-642-19893-9
DOI 10.1007/978-3-642-19893-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011922746

CR Subject Classification (1998): F.2, G.1.6, G.1.2, I.2.8

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

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

Printed on acid-free paper

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

Preface

EMO is a bi-annual international conference series devoted to evolutionary multi-criterion optimization. The first EMO conference was organized in Zurich (Switzerland) in 2001, and the other editions took place in Faro (Portugal) in 2003, Guanajuato (Mexico) in 2005, Matsushima-Sendai (Japan) in 2007 and Nantes (France) in 2009. Ouro Preto (Minas Gerais, Brazil) hosted the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011).

EMO 2011 received 83 paper submissions in total, with 251 authors (on average 3.02 authors per paper) from 26 countries. After a rigorous peer-review process involving 287 reviews in total (averaging 3.45 reviews per paper), 42 papers (50.6%) were accepted for presentation at the conference, including 11 papers accepted for the MCDM track. The authors of accepted papers were from 19 countries. EMO 2011 also featured four distinguished keynote speakers: Jyrki Wallenius (Aalto University), Singiresu S. Rao (University of Miami), Roman Słowiński (Poznań University of Technology) and Eckart Zitzler (PHBern).

The field of evolutionary multi-criterion optimization emerged, in the 1990s, as a confluence of the traditional field of multi-objective mathematical programming with the area of evolutionary computation that was emerging at that moment. Dealing with the simultaneous optimization of several criteria, multi-criterion optimization aimed to work with the concept of a set of efficient solutions (also called Pareto-optimal solutions), which represent different trade-off solutions for a given problem, considering the different objectives. Evolutionary computation, which developed powerful heuristic methods with inspiration from the natural phenomena of organization of living organisms, brought much flexibility and insight to the new field, making the EMO algorithms be recognized today as some of the most valuable and promising methods for tackling complex and diverse multi-criterion optimization problems.

The research on EMO focused, along the last two decades, on issues such as the design of efficient algorithmic methods for the approximation of efficient solutions, the problem of measuring the quality of the approximations generated by the algorithms, the hybridization with other currents of optimization, and so forth. A representative sample of this history is registered in the proceedings of the EMO conferences, which have been the forum for the first presentation of several breakthroughs, for the raising of new questions, and for the early indication of new trends within the research community of the area. For instance, the intriguing behavior of EMO algorithms in many-objective problems (problems with a large number of objective functions) was reported in the first EMO conference, in 2001, as has become a major issue in the related technical literature in this last decade. At EMO 2011, a whole section was devoted to this theme, now sketching some definitive answers to this question. This volume of EMO 2011 proceedings also has papers dealing with other fundamental questions of EMO theory, such as the development of algorithmically efficient tools for the evaluation of solution-set quality, theoretical questions related to solution archiving, and others. In addition, there are reports on the continuing effort in the development of algorithms, either for dealing with particular classes of problems or for new forms of processing the problem information. It is also noticeable that, as the field of EMO reaches maturity, the spectrum of applications grows: in this volume, almost one-third of the papers are related to EMO applications in a diversity of fields.

Finally, it should be mentioned that a continued effort of EMO conferences has been devoted to promoting the interaction with the related field of multicriteria decision making (MCDM). This reflects the growing awareness of the necessity of decision-making tools for dealing appropriately with the diversity of solutions that are typically delivered by EMO algorithms. At EMO 2011, from the 11 papers accepted for the MCDM track, 6 papers were presented in mixed sessions with related EMO papers – which indicates that the two communities are already finding common directions.

April 2011

R. Takahashi
K. Deb
E. Wanner
S. Greco

Organization

EMO 2011 was organized by Universidade Federal de Minas Gerais, in cooperation with Centro Federal de Educação Tecnológica de Minas Gerais and Universidade Federal de Ouro Preto.

Committees and Chairs

Conference Chairs	Ricardo H.C. Takahashi (UFMG, Brazil) Kalyanmoy Deb (IIT-Kanpur, India) Elizabeth F. Wanner (CEFET-MG, Brazil) Salvatore Greco (University of Catania, Italy)
Steering Committee	Carlos A.C. Coello (CINVESTAV-IPN, Mexico) David Corne (Heriot-Watt University, UK) Kalyanmoy Deb (IIT-Kanpur, India) Peter J. Fleming (University of Sheffield, UK) Carlos M. Fonseca (University of Coimbra, Portugal) Hisao Ishibuchi (Osaka Prefecture University, Japan) Joshua Knowles (University of Manchester, UK) Kaisa Miettinen (University of Jyvaskyla, Finland) J. David Schaffer (Phillips Research, USA) Lothar Thiele (ETH Zurich, Switzerland) Eckart Zitzler (PHBern, Switzerland)
Local Committee	Felipe Campelo (UFMG, Brazil) Rodrigo T.N. Cardoso (CEFET-MG, Brazil) Eduardo G. Carrano (CEFET-MG, Brazil) Anderson R. Duarte (UFOP, Brazil) Luiz H. Duczmal (UFMG, Brazil) Frederico G. Guimarães (UFMG, Brazil) Gisele L. Pappa (UFMG, Brazil) Oriane M. Neto (UFMG, Brazil)

International Program Committee

Abbass, H. (Australia)
Alba, E. (Spain)
Almeida Filho, A.T. (Brazil)
Alves, M.J. (Portugal)
Angilella, S. (Italy)
Antunes, C.H. (Portugal)
Autran Gomes, L.F. (Brazil)
Avigad, G. (Israel)
Bana e Costa, C.A. (Portugal)
Bandyopadhyay, S. (India)
Barbosa, H.J.C. (Brazil)
Basseur, M. (France)
Berry, A. (Australia)
Beume, N. (Germany)
Branke, J. (UK)
Brockhoff, D. (France)
Bui, L.T. (Australia)
Caballero, R. (Spain)
Carlsson, C. (Finland)
Carvalho, A. (Brazil)
Climaco, J. (Portugal)
Coelho, A.L.V. (Brazil)
Coello Coello, C.A. (Mexico)
Corne, D.W. (UK)
Costa, L. (Portugal)
Dabberu, M. (India)
De Smet, Y. (Belgium)
Delbem, A.C.B. (Brazil)
Dhaenens, C. (France)
Dias, L. (Portugal)
Doerner, K. (Germany)
Doumpos, M. (Greece)
Drechsler, R. (Germany)
Ehrgott, M. (New Zealand)
Emmerich, M.T. (The Netherlands)
Engelbrecht, A. (South Africa)
Everson, R. (UK)
Fernandez, E. (Mexico)
Fieldsend, J.E. (UK)
Figueira, J. (Portugal)
Fleming, P. (UK)
Fonseca, C.M. (Portugal)
Fortemps, P. (Belgium)
Galves, M.L. (Brazil)
Gandibleux, X. (France)
Gaspar-Cunha, A. (Portugal)
Geiger, M. (Germany)
Giannakoglou, K.C. (Greece)
Grimme, C. (Germany)
Hao, J.K. (France)
Haubelt, C. (Germany)
Heywood, M. (Canada)
Hughes, E.J. (UK)
Igel, C. (Germany)
Inuiguchi, M. (Japan)
Ishibuchi, H. (Japan)
Jaszkiewicz, A. (Poland)
Jin, Y. (UK)
Jones, D. (UK)
Jourdan, L. (France)
Katoh, N. (Japan)
Knowles, J.D. (UK)
Koeppen, M. (Japan)
Koksalan, M. (Turkey)
Kumar, R. (India)
Kurz, M.E. (USA)
Landa-Silva, D. (UK)
Li, X. (Australia)
Liefooghe, A. (Portugal)
Lins, M.E. (Brazil)
Lozano, J.A. (Spain)
Marti, L. (Spain)
Matarazzo, B. (Italy)
Matos, M. (Portugal)
Mello, J.C.S. (Brazil)
Mezura-Montes, E. (Mexico)
Middendorf, M. (Germany)
Miettinen, K. (Finland)
Molina, J. (Spain)
Montibeller, G. (UK)
Morton, A. (UK)
Mostaghim, S. (Germany)
Mousseau, V. (France)
Naujoks, B. (Germany)
Nebro, A. (Spain)
Ng, A. (Sweden)
Ogryczak, W. (Poland)
Okabe, T. (Japan)
Oliveira, P. (Portugal)
Padhye, N. (India)
Paquete, L. (Portugal)
Parks, G. (UK)
Parreiras, R. (Brazil)
Pelikan, M. (USA)
Poles, S. (Italy)
Pozo, A.T.R. (Brazil)
Purshouse, R.C. (UK)
Ramírez, J.A. (Brazil)
Ranjithan, R.S. (USA)
Ray, T. (Australia)
Rockett, P. (UK)
Rodríguez-Vazquez, K. (Mexico)
Romero, C. (Spain)
Roy, B. (France)
Ruiz, F. (Spain)
Runarsson, T.P. (Iceland)
Ruzika, S. (Germany)
Saha, S. (India)
Sakuma, J. (Japan)
Schmeck, H. (Germany)
Schoenauer, M. (France)
Schütze, O. (Mexico)
Sevaux, M. (France)
Siarry, P. (France)
Simões Gomes, C.F. (Brazil)
Sindhya, K. (Finland)
Sinha, A. (Finland)

Siskos, Y. (Greece)
Slowinski, R. (Poland)
Steuer, R.E. (USA)
Subbu, R. (USA)
Tan, K.C. (Singapore)
Teghem, J. (Belgium)
Teich, J. (Germany)
Thiele, L. (Switzerland)
Tiwari, A. (UK)
Tiwari, S. (USA)
Tsoukias, A. (France)
Vanderpooten, D. (France)
Vasconcelos Silva, J.S. (Brazil)
Vasconcelos, J.A. (Brazil)
Vetschera, R. (Austria)
Von Zuben, F.J. (Brazil)
Wagner, T. (Germany)
Wallenius, J. (Finland)
Wang, L. Z. (China)
Wiecek, M.M. (USA)
Yalaoui, F. (France)
Yen, G. G. (USA)
Yildiz, A.R. (USA)
Zanakis, S.H. (USA)
Zhang, Q. (UK)
Zitzler, E. (Switzerland)
Zopounidis, C. (Greece)

Support Institutions

Sociedade Brasileira de Computação (SBC)
Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)
Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG)
Companhia Energética de Minas Gerais (CEMIG)
Empresa Brasileira de Aeronáutica (EMBRAER)

Table of Contents

I Theoretical Issues

II Algorithms

III Applications

IV MCDM

Automated Innovization for Simultaneous Discovery of Multiple Rules in Bi-objective Problems

Sunith Bandaru and Kalyanmoy Deb

Indian Institute of Technology Kanpur, Kanpur, UP 208016, India
{sunithb,deb}@iitk.ac.in
http://www.iitk.ac.in/kangal

Abstract. The trade-off solutions of a multi-objective optimization problem, as a whole, often hold crucial information in the form of rules. These rules, if predominantly present in most trade-off solutions, can be considered as the characteristic features of the Pareto-optimal front. Knowledge of such features, in addition to providing better insights to the problem, enables the designer to handcraft solutions for other optimization tasks which are structurally similar to it; thus eliminating the need to actually optimize. *Innovization* is the process of extracting these so called *design rules*. This paper proposes to move a step closer towards the complete *automation* of the innovization process through a niched clustering based optimization technique. The focus is on obtaining multiple design rules in a single knowledge discovery step using the niching strategy.

Keywords: automated innovization, multiple-rule discovery, niching, row-echelon forms.

1 Introduction: A Motivating Example

One of the goals in multi-objective problem solving is to find solutions which are as close to the true Pareto-optimal front as possible with as much diversity in decision space as possible [9]. Numerous algorithms proposed over the years have been successful in achieving this with varying degrees. Considering the amount of time, resources and research effort that has gone into developing these algorithms, it is ironic that practically only a single (or utmost a few) desirable solution(s) actually get implemented in most problems. The accuracy and diversity attained with respect to other solutions can be put to good use if they can somehow be used to gain interesting knowledge about the problem.

Consider the bi-objective design problem of a two-bar truss. The configuration is shown in Fig. 1. The problem requires that the total volume V of the truss structure be minimized along with the minimization of the maximum stress S induced in either of the bars. Geometrical constraints restrict the cross-sectional areas x_1 and x_2 of the bars the dimension y. The induced stress should remain

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 1–15, 2011.

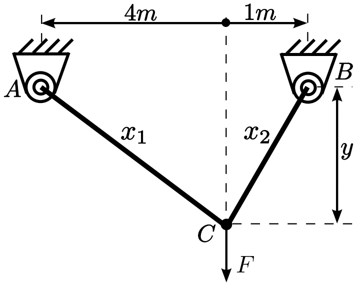


Fig. 1. Two-bar truss configuration

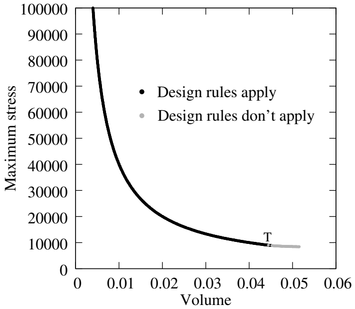


Fig. 2. Pareto-optimal front for the two-bar truss design problem

below the elastic strength S_y of the material used, which gives rise to a third constraint. The optimization formulation thus becomes,

$$\begin{array}{ll} \text{Minimize} & f_1(\mathbf{x}) = V = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}, \\ \text{Minimize} & f_2(\mathbf{x}) = S = \max(\sigma_{AC}, \sigma_{BC}), \\ \text{Subject to} & \max(\sigma_{AC}, \sigma_{BC}) \leq S_y \text{ kPa}, \\ & 0 \leq x_1, x_2 \leq 0.01 \text{ m}^2, \\ & 1 \leq y \leq 3 \text{ m}. \end{array} \tag{1}$$

It is possible to analytically derive solutions for some special multi-objective problems. Methods usually involve the use of Fritz-John conditions or Karush-Kuhn-Tucker optimality criteria and are generally used for convex multi-objective problems. For solving (1) however, the identical resource allocation strategy can be used. Increasing the cross-sectional area of one bar element reduces the stress induced in it and so the second objective takes the other bar element into account at some point. But since both the objectives are equally important, this cannot be allowed. A balance can be obtained only when the stresses in both the bars are equal. Thus,

$$\sigma_{AC} = \sigma_{BC} \Rightarrow \frac{F}{4}\frac{\sqrt{16+y^2}}{yx_1} = \frac{5F}{4}\frac{\sqrt{1+y^2}}{yx_2}. \tag{2}$$

Following a similar argument for the volumes we get,

$$x_1\sqrt{16+y^2} = x_2\sqrt{1+y^2}. \tag{3}$$

Solving (2) and (3) gives the following relationships, where the '$*$' emphasizes the fact that they represent Pareto optimality,

$$y^* = 2, \quad x_2^* = 2x_1^*, \quad V^* = 4\sqrt{5}x_1^* = 2\sqrt{5}x_2^*. \tag{4}$$

Note that these relationships will hold irrespective of the material and loading. They are thus the generic "design rules" of the truss problem in (1). Designers can remember them as guidelines when optimizing any such structure and easily handcraft a near Pareto-optimal solution. Of course the solution has to be checked for feasibility before actual implementation because the derivation of (4) did not involve the use of constraints. The gray points in Fig. 2 indicate the solutions to which these rules do not apply.

2 Related Work

The simple example discussed above shows that interesting knowledge in the form of design rules exist in multi-objective scenarios and that they are the distinguishing features of the Pareto-optimal solutions of a problem. Analytical solutions to optimization problems, especially multi-objective problems, are however rarely easy to obtain. Some alternatives exist to avoid actually solving the problems to decipher design rules. For example, monotonicity analysis [1] is capable of obtaining them directly from the problem [2] provided that the latter satisfies certain conditions of monotonicity.

In general however, there seem to be three major obstacles in deriving design rules such as those in (4): (i) most problems are not analytically solvable, (ii) methods require the problems to have a certain form or satisfy certain criteria, and (iii) other methods can only produce implicit or semantic rules. The first two of these obstacles can be overcome by employing data-mining and machine learning techniques on the solutions obtained by solving the problem through a numerical optimization algorithm. Obayashi and Sasaki [3] used self-organizing maps (SOMs) to generate clusters of design variables that indicate their role in making design improvements. A multi-objective design exploration (MODE) methodology was proposed [4] to reveal the structure of optimal design space. The study concluded that such design knowledge can be used to produce better designs. Hierarchical grouping of solutions in the form of a dendogram to identify strongly related variables is described in [5]. Recently, a data-mining technique called proper orthogonal decomposition has been used [6] to extract implicit design knowledge from the Pareto-optimal solutions.

While the above mentioned studies truly depict the importance of analyzing trade-off solutions, they fall short of providing a generic framework which can also overcome the third obstacle in discovering design rules; the ability to extract rules that are meaningful to the human designer. In other words, rules that have an explicit mathematical form are more useful and intuitive to a human for remembrance and future application to similar design problems. *Innovization* [7] addresses this, though at a simple level, through a manual graph plotting and regression analysis procedure. In the remaining sections we describe and extend the innovization procedure for complete automated discovery of multiple design rules from the Pareto-optimal solutions.

3 Discovering Design Rules through Innovization

The term innovization comes from *innov*ation through optim*ization*. As described above, it is a manual plotting-and-analysis process which can help reveal design rules hidden in Pareto-optimal datasets. Deb and Srinivasan [8] define it as "a new design methodology in the context of finding new and innovative design principles by means of optimization techniques". The basic procedure is simple and can be accomplished through the following steps:

1. Obtain the Pareto-optimal front for the multi-objective problem using any of the available population-based evolutionary algorithms [9]. Local search methods may be used to improve the obtained front [8].
2. Form the data-set containing the Pareto-optimal variable values ($\mathbf{x}^*$), objective function values ($\mathbf{f}^*$) and corresponding values of any other function(s) $\phi_j(\mathbf{x}^*)$ (see below) for all the obtained trade-off solutions.
3. Consider various entity combinations: variable-variable, objective-objective, objective-variable, etc. and plot the corresponding values for all the trade-off solutions. This will visually reveal the existence of design rules when any of these combinations show a high correlation between the entities considered. Other functions (ϕ_j's) which the designer feels are significant to the design task may also be considered.
4. In case a non-linear correlation is observed, techniques like logarithmic transformation and curve-fitting are employed to come up with the closest rule.

While steps (1) and (2) can be easily integrated into a computer code, it is the human judgement required in step (3) that makes innovization a tedious and time consuming approach to rule-finding. Nevertheless the process has been applied as such to various multi-objective problems [7,8,10] and even researchers in fields as diverse as architecture [11], virtual reality [12], robotics [13], etc. are realizing the need to identify the commonalities between solutions in the form of usable rules. Although in most innovization studies, solutions from the entire range of the Pareto-optimal front are considered, innovization from a partial set or multiple fronts from different values of problem parameters is also possible.

The truss design problem introduced previously can be used to illustrate the innovization approach. The required trade-off data is generated by solving (1) using NSGA-II [14]. A population size of 1000 is used for the purpose so as to generate a well-represented trade-off front. F is taken to be 100 kN and S_y as 10^5 kPa. Figures 3 and 4 clearly show that V-x_1, x_2-x_1 and V-x_2 have a high

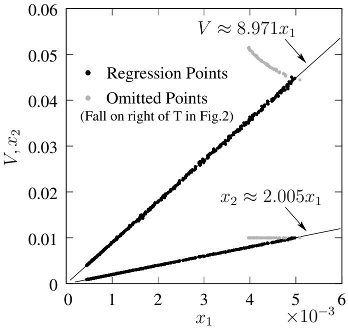


Fig. 3. Innovization for V-x_1 and x_2-x_1

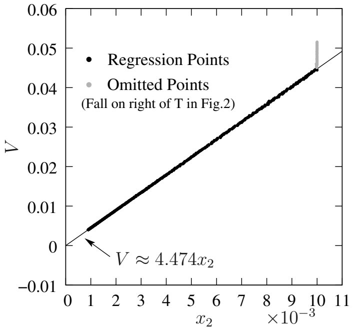


Fig. 4. Innovization for V-x_2

correlation. The slopes of regression lines fitted to each of these combinations (by ignoring points which apparently show a change in relationship) gives the constant of proportionality. Note that,

$$V \approx 8.971 x_1 \approx 4\sqrt{5} x_1, \quad x_2 \approx 2.005 x_1 \approx 2 x_1 \text{ and } \quad V = 4.474 x_2 \approx 2\sqrt{5} x_2. \quad (5)$$

As it can be seen, a post-optimality procedure like innovization is capable of revealing design rules from the trade-off solutions. However, there are some obvious difficulties which prompt an automation of this approach. Firstly, as discussed above, manually choosing different combinations and checking for correlations is a time consuming process. Secondly, all solutions need not follow a particular design rule. Unlike in Figs. 3 and 4, if the change in relationship is subtle, a blind regression analysis may lead to erroneous conclusions. Each design rule can thus be associated with a prominence level or *significance* depending on the percentage of the Pareto-optimal front that it applies to. It is crucial that rules with a low significance be filtered out by the automated algorithm. Lastly, there can be design rules which have different proportionality constants in different regions of the Pareto-optimal front [15,16]. The design rule is then said to be *parametrically* varying across these regions. The automated algorithm should be able to tell apart solutions which parametrically satisfy the rule from the ones that do not satisfy it at all.

4 Proposed Automated Innovization Approach

Let $\phi_j(\mathbf{x})$ be the set of "basis functions" whose combinations are to be analyzed for the presence of design rules. The designer can specify N such functions. They also include the variables, objectives and constraints of the problem. Previous manual innovization studies [7,8,10] and recent proof-of-principle studies towards a possible automated innovization [15,16] sufficiently show that most design rules take the generic form,

$$\Pi_{j=1}^{N} \phi_j(\mathbf{x})^{a_{ij} B_{ij}} = C_i \quad (6)$$

where C_i is the proportionality constant for the i-th design rule and B_{ij}'s are corresponding powers of the basis functions in that design rule. The Boolean variables a_{ij}'s, in a way, reflect the presence ($a_{ij} = 1$) or absence ($a_{ij} = 0$) of the j-th basis function in the i-th rule. In addition to being relatively easier for an automated algorithm to detect, the mathematical form of these relationships makes them more intuitive to the user. Note that B_{ij}'s in (6) can take any real value depending on the problem. In order to restrict the search space, the maximum absolute power (among participating basis functions) is set to one by making the following transformation which keeps the design rule unaltered:

$$\left(\Pi_{j=1}^{N} \phi_j(\mathbf{x})^{a_{ij} B_{ij}}\right)^{\overline{\{B_{ip} | p : (\max_p |a_{ip} B_{ip}|)\}}^{\,1}} = C_i^{\overline{\{B_{ip} | p : (\max_p |a_{ip} B_{ip}|)\}}^{\,1}}$$
$$= \quad c_i \quad \text{(say)}, \quad (7)$$

which can be simply written as,

$$\Pi_{j=1}^{N}\phi_j(\mathbf{x})^{a_{ij}b_{ij}} = c_i, \quad \text{where } b_{ij} = \frac{B_{ij}}{\{B_{ip}|p:(\max_p|a_{ip}B_{ip}|)\}} \in [-1,1]. \tag{8}$$

Supposing that a_{ij}'s and b_{ij}'s are known for the i-th rule, the parameter c_i can easily be calculated for all trade-off solutions in the Pareto-optimal dataset. Solutions that parametrically satisfy the i-th rule will have the same (or nearly equal) values for c_i. The present algorithm uses a grid-based clustering technique to identify clusters of such solutions. If a solution yields a c_i value which is significantly different from that of others, then it remains unclustered.

4.1 Finding Optimal a_{ij}'s and b_{ij}'s

Each cluster mentioned above corresponds to one region of the Pareto-optimal front to which the i-th design rule applies parametrically. The c_i-values in each of these clusters should therefore be nearly equal. In other words, the spread of c_i-values in each cluster should be minimum. This condition can be used to obtain a_{ij}'s and b_{ij}'s. To ensure that a *narrow* distribution of c_i-values is obtained in the clusters, the coefficient of variation (c_v= standard deviation/mean) of the c_i-values is simultaneously minimized in all of them. The weighted sum approach is used to have a computationally tractable approach. Since all clusters are equally important for the design rule to be valid, the c_v's are assigned equal weights. Note that c_v being a *normalized measure of spread* will have the same order of magnitude in all the clusters and therefore the weighted sum approach should suffice [16]. Thus, to find optimal a_{ij}'s and b_{ij}'s for the i-th rule the following optimization problem can be solved,

$$\begin{array}{ll} \text{Minimize} & \sum\limits_{clusters} c_v^{(k)}, \qquad c_v^{(k)} = \dfrac{\sigma_{c_i}}{\mu_{c_i}} \quad \forall\, c_i \in k\text{-th cluster} \\ \text{Subject to} & -1.0 \le b_{ij} \le 1.0 \quad \forall\, j : a_{ij} = 1, \\ & |b_{ij}| \ge 0.1 \quad \forall\, j : a_{ij} = 1, \\ & \sum_j a_{ij} \ge 1, \\ & a_{ij}\text{'s are Boolean and } b_{ij}\text{'s are real.} \end{array} \tag{9}$$

Here c_i is obtained from (8). Any basis function ϕ_j with a low magnitude power b_{ij} will hardly contribute to the design rule. Moreover, inclusion of zeroes in the search space will lead to a trivial solution where $c_i = 1$ for all trade-off solutions and so $c_v = 0$ for all clusters. The second set of constraints checks this by putting a lower bound on the magnitude of b_{ij} (see Sect. 5.3). The last constraint ensures that at least one basis function is used to form the design rule.

4.2 One-Dimensional Grid-Based Clustering

Grid-based clustering technique involves partitioning the data into a number of grids (or divisions) and merging them to form clusters. The number of clusters need not be predefined. The following steps describe the procedure:

Step 1: Sort all $\{c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(m)}\}$ obtained by evaluating (8) for all m trade-off solutions in the data-set.

Step 2: Divide the space $[c_{i,min}, c_{i,max}]$ into, say d_i divisions.

Step 3: Count the number of points in each division.

Step 4: Merge adjacent divisions which have more than or same as $\lfloor \frac{m}{d_i} \rfloor$ points (the average number of points per division) to form clusters.

Step 5: Count the number of clusters $\mathcal{C}_i$ and the total number of unclustered points $\mathcal{U}_i$ in all divisions with less than $\lfloor \frac{m}{d_i} \rfloor$ points.

There are two conflicting arguments for deciding the number of divisions d_i. It is desirable that the design rule be applicable to as many points as possible, so ideally $\mathcal{U}_i = 0$. This translates to a high value of d_i since each unclustered point can then form a one-element cluster. But this, in turn will increase the number of clusters. A good clustering algorithm should be able to find the simplest representation of the data (which points to a low d_i) while ensuring that points within a cluster are similar in some respect. In the present case, c_v's within the clusters define this similarity. Thus, finding the optimum number of divisions can be framed as an optimization problem,

$$\begin{aligned} &\text{Minimize} && \mathcal{C}_i + \sum_{k=1}^{\mathcal{C}_i} c_v^{(k)} \times 100\%, \qquad c_v^{(k)} = \frac{\sigma_{c_i}}{\mu_{c_i}} \quad \forall\, c_i \in k\text{-th cluster}, \\ &\text{Subject to} && \mathcal{U}_i = 0, \\ &&& 1 \le d_i \le m, \\ &&& d_i \text{ is an integer.} \end{aligned} \tag{10}$$

The limits on d_i are due to the clustering criterion of $\lfloor \frac{m}{d_i} \rfloor$ points. It is easy to see that any value of $d_i > m$ would yield the same result as $d_i = m$. The percentage coefficient of variation is used to approximately scale the c_v-values to the order of number of clusters ($\mathcal{C}_i$) which in turn allows the use of the weighted sum [16].

The clustering algorithm discussed above uses the a_{ij} and b_{ij} values obtained from the methodology described in Sect. 4.1 to calculate d_i. The latter in turn uses the clusters identified by the former to calculate the c_v's. The two optimization problems (9) and (10) therefore have to be solved simultaneously. The algorithm calculations involved in both of these and the non-availability of mathematical functions prevents the use of a classical optimization approach. The present paper uses a simple genetic algorithm (GA) instead. It has the added advantage that a_{ij}'s can simply be coded as the bits of a binary string whereas b_{ij}'s can be regarded as real variables. Each population member then acts as a design rule and only the best among these survive.

4.3 Significance of Design Rules

As discussed towards the end of Sect. 3, the discovered design rules should be significant for them to be useful for a designer. A direct measure of significance is the percentage of trade-off data-set that the rule applies to. To calculate the significance of each GA population member, the fourth step of the clustering algorithm in Sect. 4.2 is modified as,

Modified Step 4: Merge adjacent divisions which have more than or same as $\lfloor \frac{m}{d_i} \rfloor + \epsilon$ points to form clusters.

With a small integer value for ϵ, divisions which barely form part of the clusters can be identified. Let $\mathcal{C}_{i,MS}$ and $\mathcal{U}_{i,MS}$ respectively be the number of clusters and unclustered points calculated with the modified step. Note that the clustering itself need not be redone for this purpose. The significance S_i of the i-th design rule can now be given as,

$$S_i = \frac{m - \mathcal{U}_{i,MS}}{m} \times 100\%. \tag{11}$$

By placing a lower bound on S_i (say $S_{reqd} = 80\%$), designers can choose the minimum level of prominence for the design rules.

4.4 Niching for Multiple Design Rules

The discussion so far has been carried out with respect to the i-th design rule. However, solving a simple superposition of problems (9) and (10) with an additional constraint on significance (11) will only yield a single design rule because only the best population member will survive through the generations. The co-existence of multiple design rules can be promoted by introducing a *niched-tournament* selection operator [17] in the GA which allows a tournament to be played only between population members which use the same set of basis functions. The niching is implemented on top of the penalty-parameter-less approach to constraint handling [18]. The following criteria are used to determine the winner among two solutions u and v participating in a tournament:

1. If $a_{uj} = a_{vj} \quad \forall\, j = 1, 2, \ldots, N$, then u and v can be compared.
   (a) If one is feasible and the other is not then the feasible solution is preferred.
   (b) If both are feasible then the one with better objective value is preferred.
   (c) If both are infeasible then the one with lower constraint violation is preferred.
2. Else both u and v are competent.

With a GA that uses (i) this new niched-tournament selection operator for handling the constraints, (ii) one-point crossover and bit-wise mutation for the binary string of a_{ij} bits, (iii) simulated binary crossover (SBX) and polynomial mutation for b_{ij}'s and, (iv) a discrete version of SBX and polynomial mutation for the variable d_i, the combined optimization problem to be solved for extracting multiple design rules simultaneously is proposed as,

$$\begin{aligned}
&\text{Minimize} && \mathcal{C}_i + \sum_{k=1}^{\mathcal{C}_i} c_v^{(k)} \times 100\%, \qquad c_v^{(k)} = \frac{\sigma_{c_i}}{\mu_{c_i}} \quad \forall\, c_i \in k\text{-th cluster} \\
&\text{Subject to} && -1.0 \le b_{ij} \le 1.0 \quad \forall\, j : a_{ij} = 1, \\
&&& |b_{ij}| \ge 0.1 \quad \forall\, j : a_{ij} = 1, \\
&&& \textstyle\sum_j a_{ij} \ge 1, \\
&&& \mathcal{U}_i = 0, \quad 1 \le d_i \le m, \quad S_i \ge S_{reqd}, \\
&&& a_{ij}\text{'s are Boolean, } b_{ij}\text{'s are real and } d_i \text{ is an integer.}
\end{aligned} \tag{12}$$

5 Results

Algorithm 1 summarizes the proposed automated innovization approach for discovering multiple design rules in a single knowledge discovery step. It is now applied in this form to two well-studied engineering design problems.

Algorithm 1. Automated innovization for simultaneous multiple rule discovery.

```
Obtain a good set of $m$ diverse and near Pareto-optimal solutions for the multi-objective problem.
Choose the terminal set of $N$ basis functions $\phi_j$'s.
Form the $m \times N$ data-set of $\phi_j$'s evaluated at all $m$ trade-off solutions.
Initialize variables $a_{ij}$, $b_{ij}$ and $d_i$ $\forall$ $i = popsize$ GA members.
$gen \leftarrow 1$
while $gen \leq maxgen$ do
  for $i := 1$ to $popsize$ do
    Transform $b_{ij} \leftarrow \frac{b_{ij}}{\{b_{ip} | p : (\max_p |a_{ip} b_{ip}|)\}}$ to maintain $\max |a_{ij} b_{ij}| = 1$
    Evaluate $c_i = \Pi_{j=1}^{N} \phi_j(\mathbf{x})^{a_{ij} b_{ij}}$ $\forall$ $m$ trade-off solutions.
    Sort and cluster $\{c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(m)}\}$ using the grid-based clustering.
    Evaluate the objective and constraints in (12).
  end for
  Perform niched-tournament selection on the GA population members.
  Perform appropriate crossover operations.
  Perform appropriate mutation operations.
  Update GA population members.
  $gen \leftarrow gen + 1$
end while
Report unique members of final GA population as the obtained design rules ($\mathbf{D}$).
```

5.1 Truss Design Revisited

The trade-off solutions obtained in Sect. 3 for $m = 1,000$ points using NSGA-II are utilized here. All objectives and variables are chosen as the basis functions. Hence,

$$\phi_1 = V, \quad \phi_2 = S, \quad \phi_3 = x_1, \quad \phi_4 = x_2, \quad \phi_5 = y.$$

The following parameters are used for solving (12):

Population Size ($popsize$) = 400
Maximum number of generation ($maxgen$) = 500
One-point crossover probability ($p_{c,binary}$) = 0.85
Bit-wise mutation probability ($p_{m,binary}$) = 0.15
Continuous and discrete SBX probability ($p_{c,real}$) = 0.95
Continuous and discrete polynomial mutation probability ($p_{m,real}$) = 0.05
Continuous and discrete SBX distribution index (η_c) = 10
Continuous and discrete polynomial mutation distribution index (η_m) = 50
Parameter for calculating $\mathcal{U}_{i,MS}$ (ϵ) = 3
Threshold significance for design rules (S_{reqd}) = 80%

Table 1 shows the *unique* solutions present in the GA population after 500 generations of the proposed algorithm. Notice how the use of niched-tournament selection and the constraint on significance together helped maintain diverse (in terms of basis functions) and yet significant design rules. For example, the $i = 20$-th design rule is as follows:

$$V^{1.0000000}S^{0.7911332}x_1^{-0.2102126} = c_{20}. \tag{13}$$

The essence of this design rule is not in the parametric constant c_{20} but in the fact that the left-hand side of this expression remains almost constant for at least $S_{reqd} = 80\%$ of the NSGA-II obtained data-set.

Table 1. Design rules (**D**) obtained for the truss design problem

i	$a^*_{ij}\ \forall\ j$	$a^*_{i1}b^*_{i1}$	$a^*_{i2}b^*_{i2}$	$a^*_{i3}b^*_{i3}$	$a^*_{i4}b^*_{i4}$	$a^*_{i5}b^*_{i5}$	d^*_i
1	10010	-0.9979552	0.0000000	0.0000000	1.0000000	0.0000000	869
2	01111	0.0000000	-0.9668656	-0.7030048	-0.2639445	1.0000000	869
3	01011	0.0000000	-0.7727935	0.0000000	-0.7749417	1.0000000	869
4	01101	0.0000000	-0.7048297	-0.7049160	0.0000000	1.0000000	870
5	00110	0.0000000	0.0000000	-0.9990348	1.0000000	0.0000000	869
6	00111	0.0000000	0.0000000	-0.9952743	0.9999844	1.0000000	869
7	00001	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	869
8	11111	0.7817370	-0.7754779	-0.7850468	-0.7734357	1.0000000	869
9	10011	0.8268434	0.0000000	0.0000000	-0.8259557	1.0000000	869
10	10101	0.8388214	0.0000000	-0.8391667	0.0000000	1.0000000	869
11	10111	0.8813441	0.0000000	-0.5827007	-0.3013669	1.0000000	869
12	11101	0.9486451	0.1086554	-0.8411865	0.0000000	1.0000000	856
13	11011	0.9962421	0.6705470	0.0000000	-0.3253438	1.0000000	908
14	11001	0.9989458	0.9984587	0.0000000	0.0000000	1.0000000	869
15	11000	0.9999623	1.0000000	0.0000000	0.0000000	0.0000000	869
16	11110	1.0000000	-0.5590224	-0.7850468	-0.7741561	0.0000000	869
17	10100	1.0000000	0.0000000	-0.9971116	0.0000000	0.0000000	869
18	10110	1.0000000	0.0000000	-0.7353538	-0.2647701	0.0000000	869
19	11010	1.0000000	0.6702390	0.0000000	-0.3300355	0.0000000	869
20	11100	1.0000000	0.7911332	-0.2102126	0.0000000	0.0000000	860

There is, however, a downside to discovering and presenting design rules in the form as in Table 1. A human designer would prefer having all the design rules in a compact form which can be intuitive to him/her. It is not difficult to see that there are some *redundant* rules in Table 1. The next logical step is therefore to condense and present them in a more compact form.

5.2 Reduced Row-Echelon Form

In linear algebra, reduced row-echelon forms (RREF) are used to identify linearly dependent rows of a matrix. By eliminating redundant variables using RREF, a system of linear equations can be solved easily. Here, the same technique is used to condense the design rules. However, since the original trade-off data-set is only near Pareto-optimal, the obtained design rules can only be approximate. Therefore, a tolerance *tol* should be allowed during row operations. Algorithm 2 is used for this purpose.

Algorithm 2. Determination of *tol* and condensed design rules.

```
$l \leftarrow$ maximum number of significant digits in any element of $\mathbf{D}$.
repeat
    $tol \leftarrow 10^{-l}$
    $\mathbf{D}_{reduced} = \mathrm{rref}(\mathbf{D}, tol)$    {The MATLAB® function rref() is used here.}
    $l \leftarrow l - 1$
until rank $\mathbf{D}_{reduced} < N$
Identify insignificant relationships in $\mathbf{D}_{reduced}$ by performing grid-based clustering.
Report other relationships as the condensed design rules.
```

Table 2. Reduced design rules ($\mathbf{D}_{reduced}$) for the truss design problem, $tol = 0.01$

i	$a^*_{i1}b^*_{i1}$	$a^*_{i2}b^*_{i2}$	$a^*_{i3}b^*_{i3}$	$a^*_{i4}b^*_{i4}$	$a^*_{i5}b^*_{i5}$	d^*_i	S_i
DR1	1.0000000	0.0000000	0.0000000	-1.0006158	0.0000000	520	88.2%
DR2	0.0000000	1.0000000	0.0000000	1.0005781	0.0000000	508	80.8%
DR3	0.0000000	0.0000000	1.0000000	-1.0009661	0.0000000	507	86.8%
DR4	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	511	87.2%

Table 2 shows the result of applying Algorithm 2 on the values in Table 1. It can be seen that all the original rules in (4) are realizable from these condensed rules whose approximate forms are summarized below:

$$\text{DR1: } \frac{V}{x_1} = c_{\text{DR1}}, \quad \text{DR2: } Sx_2 = c_{\text{DR2}}, \quad \text{DR3: } \frac{x_1}{x_2} = c_{\text{DR3}}, \quad \text{DR4: } y = c_{\text{DR4}}.$$

Further insight can be obtained by again performing grid-based clustering on the c_i-values of these four rules to determine the number of clusters ($\mathcal{C}_{i,MS}$) and unclustered points ($\mathcal{U}_{i,MS}$). The corresponding d^*_i's and significance S_i are also shown in the table. Figures 5, 6, 7 and 8 show the distribution of c_i's, the horizontal broken lines being their cluster averages. Unclustered points are shown in gray.

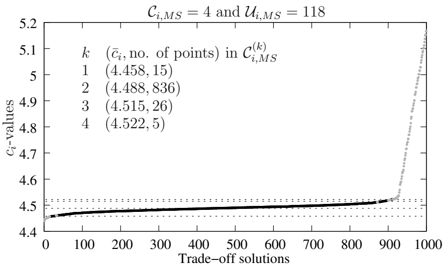


Fig. 5. Design rule i = DR1 ($S_i = 88.2\%$) is equivalent to $V = 2\sqrt{5}x_2 = 4.472x_2$

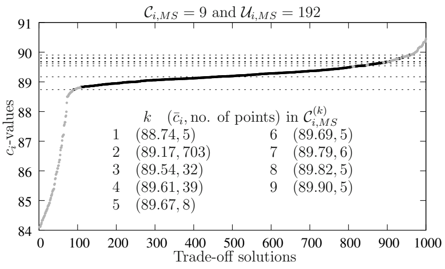


Fig. 6. Design rule i = DR2 ($S_i = 80.8\%$) is equivalent to $Sx_2 = 200/\sqrt{5} = 89.44$

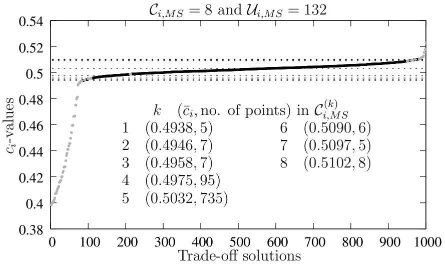


Fig. 7. Design rule $i =$ DR3 ($S_i = 86.8\%$) is equivalent to $x_2 = 2x_1$ or $x_1 = 0.5x_2$

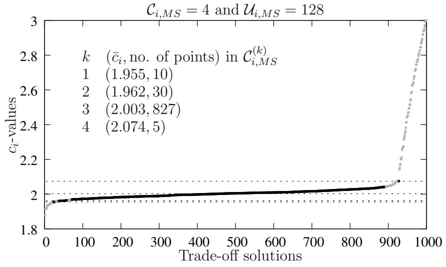


Fig. 8. Design rule $i =$ DR4 ($S_i = 87.2\%$) is equivalent to $y = 2$

The figures and the corresponding relationships show how the proposed automated algorithm is capable of successfully deciphering important rules for a design problem directly from the trade-off data-set. Specifically, for this problem it can be seen that indeed the obtained rules and the associated c_i's are approximately similar to those derived theoretically (and manually) earlier in ([4]). In a complex design scenario, such rules generated automatically will be extremely useful for the designer.

5.3 Welded Beam Design

This problem involves the minimization of cost (C) and end deflection (D) of a welded cantilever beam carrying a given maximum load. The design variables are the thickness of the beam b, width of the beam t, length of the weld l and weld thickness h. Constraints are used to limit the allowable bending stress (σ), shear stress (τ) and buckling force (P_c). The multi-objective formulation can be found in [18]. The trade-off front is obtained for $m = 300$ population size using NSGA-II. Next, the proposed algorithm is used on this data-set with $popsize = 600$ and $maxgen = 800$. All other parameters are same as in the truss design problem. The following basis functions are considered:

$$\phi_1 = C, \quad \phi_2 = D, \quad \phi_3 = b, \quad \phi_4 = t, \quad \phi_5 = l, \quad \phi_6 = h \quad \phi_7 = \sigma \quad \phi_8 = P_c.$$

The **D** matrix contains 213 unique design rules which reduce to just the seven relationships shown in Table 3. The following relations which were shown in a

Table 3. Reduced design rules ($\mathbf{D}_{reduced}$) for the welded beam problem, $tol = 0.01$

i	$a^*_{i1}b^*_{i1}$	$a^*_{i2}b^*_{i2}$	$a^*_{i3}b^*_{i3}$	$a^*_{i4}b^*_{i4}$	$a^*_{i5}b^*_{i5}$	$a^*_{i6}b^*_{i6}$	$a^*_{i7}b^*_{i7}$	$a^*_{i8}b^*_{i8}$	d^*_i	S_i
DR1	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	-0.2983906	152	34.7%
DR2	0.00000	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.3334565	154	86.0%
DR3	0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	0.00000	-0.3328624	157	88.0%
DR4	0.00000	0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	-0.0000215	155	83.3%
DR5	0.00000	0.00000	0.00000	0.00000	1.00000	0.00000	0.00000	0.2031956	158	34.0%
DR6	0.00000	0.00000	0.00000	0.00000	0.00000	1.00000	0.00000	-0.1844011	151	49.3%
DR7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.00000	0.3334762	167	84.0%

previous innovization study [8] to be the design rules of this problem are the approximate forms of these automatically obtained condensed rules:

$$\text{DR2: } DP_c^{0.333} = c_{\text{DR2}}, \text{DR3: } \frac{b}{P_c^{0.333}} = c_{\text{DR3}}, \text{DR4: } t = c_{\text{DR4}}, \text{DR7: } \sigma P_c^{0.333} = c_{\text{DR7}}.$$

The original study also shows other design rules which can be derived from the above four rules. For example, DR2 and DR3 give $Db = constant$. Similarly, DR2 and DR7 together suggest that $D \propto \sigma$ and so on.

These rules can be very handy for a designer. As an example, consider DR2 which indicates that the beam deflection $D \propto \frac{1}{P_c^{0.333}}$, the proportionality constant being $\bar{c}_i = 0.1704$ as shown in Fig. 9. A designer can convert a similar bi-objective problem into a single objective problem and still be able to create most of the trade-off solutions simply by using this rule. Similarly, the constraint on the σ can be safely eliminated by noting that $\sigma = \frac{391208.7}{P_c^{0.333}}$ applies to 252 out of the 300 trade-off solutions as shown in Fig. 10. To check whether the two design rules are applicable to the same region of the trade-off front, the clustered and unclustered points from both can be mapped to the front as shown in the insets of the two figures.

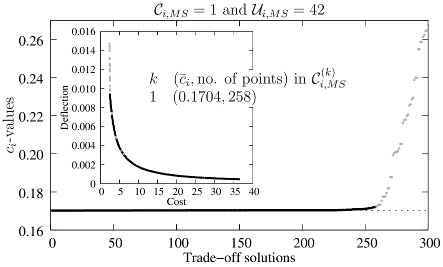


Fig. 9. Design rule i = DR2 and the corresponding mapping of points on the Pareto-optimal front

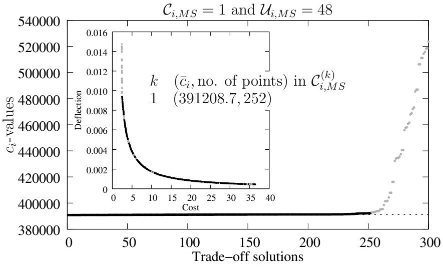


Fig. 10. Design rule i = DR7 and the corresponding mapping of points on the Pareto-optimal front

Table 3 also shows three relations that do not satisfy the criterion of being applicable to at least 80% of the data-set, namely, DR1, DR5 and DR6. They occur despite the constraint on significance because of two reasons: (i) the constraint on the magnitudes of b_{ij}'s in (12) sometimes causes spurious (yet significant) relationships to creep into the $\mathbf{D}$ matrix which carry on into $\mathbf{D}_{reduced}$ but lose their significance during the row-echelon transformation. Fortunately, they can be identified by again performing grid-based clustering as done here. In fact, the above problem was solved with 0.01 instead of 0.1 as the lower bound on magnitude since the latter resulted in DR3 having a significance of only 46% with

$a^*_{i8}b^*_{i8} = -0.3314016$. This difficulty can be alleviated, if a different lower bound can be set adaptively for each $|b_{ij}|$. (ii) the reduction to echelon form causes accumulation of the errors in b_{ij}'s which in turn are due to the near Pareto-optimal nature of the data-set. While the latter can be controlled by ensuring proximity to the Pareto-optimal front using local search methods, work is in progress at Kanpur Genetic Algorithms Laboratory (KanGAL) to take care of the former.

6 Conclusions and Future Work

The authors' earlier work [16] on automatically deciphering design principles or rules requires that the user choose various combinations of basis functions; thus discovering each relationship one at a time. The proposed algorithm addresses the problem of finding multiple such rules simultaneously in a single optimization run by automatically eliminating unwanted basis functions from the provided set. It integrates a clustering based optimization approach with a niched-tournament selection operator to allow multiple design rules to co-exist in a GA population. Additional constraints are used to discourage insignificant rules. The algorithm is demonstrated on the truss and welded beam design problems successfully and reveals interesting information about the Pareto-optimal solutions in both cases. Intuitive and easy-to-use design rules are obtained by condensing the original rules using the reduced row-echelon form. A systematic procedure for determining the tolerance required during row operations is also developed. Using a well-optimized Pareto-optimal data-set, the algorithm, as a whole, provides compact design rules which can be easily stored and retrieved for future design tasks of similar nature.

This study can be extended in a number of ways towards achieving a complete automation of the innovization process. Firstly, as seen in the welded beam problem, constraining the magnitude of b_{ij} values to a lower bound caused certain unwanted relationships to appear as rules. Though they could be identified after a row-echelon reduction and subsequent grid-based clustering, the original algorithm itself can be modified to adaptively change this lower bound. One likely approach is to set it to a value below which the corresponding ϕ_j is incapable of amounting to a threshold variation in c_i-values.

Another extension of the algorithm can be made for discrete variable problems. The limited values which a variable can take in such problems may drive the present algorithm towards trivial design rules (such as, $x_i = constant$), each satisfying a significant fraction of available data. If all the variables are discrete, the row-echelon form may never have a rank less than N with a reasonable tolerance. Further studies are needed to modify the current algorithm.

Rules in combinatorial optimization problems may be different than the ones obtained here. It will be an interesting study to modify the current algorithm for such problems as well. It would also be an interesting task to extend the proposed approach in more than two-objective problems.

References

1. Papalambros, P.Y., Wilde, D.J.: Principles of optimal design: Modeling and computation. Cambridge University Press, Cambridge (2000)
2. Deb, K., Srinivasan, A.: Monotonicity analysis, discovery of design principles, and theoretically accurate evolutionary multi-objective optimization. Journal of Universal Computer Science 13(7), 955–970 (2007)
3. Obayashi, S., Sasaki, D.: Visualization and data mining of pareto solutions using self-organizing map. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 796–809. Springer, Heidelberg (2003)
4. Obayashi, S., Jeong, S., Chiba, K.: Multi-objective design exploration for aerodynamic configurations. In: Proceedings of 35th AIAA Fluid Dynamics Conference and Exhibit, AIAA 2005-4666 (2005)
5. Ulrich, T., Brockhoff, D., Zitzler, E.: Pattern identification in Pareto-set approximations. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 737–744. ACM, New York (2008)
6. Oyama, A., Nonomura, T., Fujii, K.: Data mining of Pareto-optimal transonic airfoil shapes using proper orthogonal decomposition. In: AIAA2009-4000. AIAA, Reston (2009)
7. Deb, K.: Unveiling innovative design principles by means of multiple conflicting objectives. Engineering Optimization 35(5), 445–470 (2003)
8. Deb, K., Srinivasan, A.: Innovization: Innovating design principles through optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006, pp. 1629–1636. ACM, New York (2006)
9. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, New York (2001)
10. Datta, D., Deb, K., Fonseca, C.: Multi-objective evolutionary algorithms for resource allocation problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 401–416. Springer, Heidelberg (2007)
11. Bittermann, M.: Personal communication (July 2010)
12. Madetoja, E., Ruotsalainen, H., Mönkkönen, V.: New visualization aspects related to intelligent solution procedure in papermaking optimization. In: EngOpt 2008 - International Conference on Engineering Optimization (2008)
13. Doncieux, S., Mouret, J., Bredeche, N.: Exploring new horizons in evolutionary design of robots. In: Workshop on Exploring new horizons in Evolutionary Design of Robots at IROS, pp. 5–12 (2009)
14. Deb, K., Agarwal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
15. Bandaru, S., Deb, K.: Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design. In: IEEE Congress on Evolutionary Computation (CEC-2010), pp. 1224–1231. IEEE Press, Los Alamitos (2010)
16. Bandaru, S., Deb, K.: Towards automating the discovery of certain innovative design principles through a clustering based optimization technique. Engineering Optimization (in press), http://www.iitk.ac.in/kangal/papers/k2010001.pdf
17. Oei, C.K., Goldberg, D.E., Chang, S.J.: Tournament selection, niching, and the preservation of diversity. IlliGAL Report No. 91011, Urbana, IL: University of Illinois at Urbana-Champaign (1991)
18. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)

A Taxonomy of Online Stopping Criteria for Multi-Objective Evolutionary Algorithms

Tobias Wagner[1], Heike Trautmann[2], and Luis Martí[3]

[1] Institute of Machining Technology (ISF), TU Dortmund
Baroper Straße 301, 44227 Dortmund, Germany
wagner@isf.de
http://www.isf.de

[2] Department of Computational Statistics, TU Dortmund
Vogelpothsweg 87, 44227 Dortmund, Germany
trautmann@statistik.tu-dortmund.de
http://www.statistik.tu-dortmund.de

[3] Group of Applied Artificial Intelligence, Universidad Carlos III de Madrid
Av. de la Universidad Carlos III 22, 28270 Colmenarejo, Madrid, Spain
lmarti@inf.uc3m.es
http://www.giaa.inf.uc3m.es/

Abstract. The use of multi-objective evolutionary algorithms for solving black-box problems with multiple conflicting objectives has become an important research area. However, when no gradient information is available, the examination of formal convergence or optimality criteria is often impossible. Thus, sophisticated heuristic online stopping criteria (OSC) have recently become subject of intensive research. In order to establish formal guidelines for a systematic research, we present a taxonomy of OSC in this paper. We integrate the known approaches within the taxonomy and discuss them by extracting their building blocks. The formal structure of the taxonomy is used as a basis for the implementation of a comprehensive MATLAB toolbox. Both contributions, the formal taxonomy and the MATLAB implementation, provide a framework for the analysis and evaluation of existing and new OSC approaches.

Keywords: Convergence Detection, Multi-Objective Optimization, Performance Indicators, Performance Assessment, Termination Criterion.

1 Introduction

In recent years, the use of evolutionary algorithms (EAs) for solving multi-objective optimization problems has become established. The search for a set of solutions which approximates the Pareto-optimal front of a problem corresponds well to the population maintained within an EA. In particular for black-box problems where no gradient information is available, the use of biologically-inspired stochastic variation and selection operators provides a successful alternative.

However, without gradient information, the examination of formal convergence or optimality criteria, e. g., the Karush-Kuhn-Tucker conditions, is impossible. Therefore, the termination of multi-objective EA (MOEA) is often decided

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 16–30, 2011.

based on heuristic stopping criteria, such as the maximum number of evaluations or a desired value of a performance indicator. Whereas the criteria are suitable for analytically defined benchmark problems, where the optimal indicator value is known, their applicability to real-world black-box problems is questionable. In cases where the evaluation budget or the desired indicator level is inappropriately specified, the MOEA can either waste computational resources or can be stopped although the approximation still shows a significant improvement. Consequently, heuristic stopping criteria for the online detection of the generation, where the expected improvement in the approximation quality does not justify the costs of additional evaluations, provide an important contribution to the efficiency of MOEA.

In line with these findings, research on sophisticated heuristic online stopping criteria (OSC) has obtained increasing popularity (e.g. [8,12,16,19,20]). OSC analyze the progression of single or multiple progress indicators (PI) online during the run of the MOEA. When the considered indicators seem to be converged, i. e., the expected improvement seems to be lower than a predefined threshold, the MOEA is terminated in order to avoid wasting computational resources.

Despite being proposed by different authors with different methodological background, all these criteria show structural similarities. Thus, a taxonomy of OSC is presented which is the formal contribution of this paper and makes up the basis for the implementation of the MATLAB toolbox. Based on the foundations of set-based multi-objective optimization (section 2), the special requirements for multi-objective OSC are identified. In the main section 3, a formal framework is defined (3.1), known OSC approaches are integrated within the taxonomy by structuring them into their building blocks (3.2), and a discussion of the state of the art is provided (3.3). All building blocks identified in the taxonomy are made available in a MATLAB toolbox [18] which is briefly described in section 4. By means of this toolbox, all existing and many new OSC can be designed and analyzed. The paper is summarized and conclusions are given in section 5.

2 Foundations

Without loss of generality[1], a MOP can be formally expressed as

$$\min_{\boldsymbol{x}\in\mathcal{D}} \boldsymbol{f}(\boldsymbol{x}) = \langle f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\rangle \,, \tag{1}$$

i. e., a vector of objective functions $f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ is jointly optimized. The feasible region $\mathcal{D} \subseteq \mathcal{X}$ of the search space $\mathcal{X}$ is denoted as decision space while the image set $\mathcal{O} \subseteq \mathbb{R}^m$ of the projection $\boldsymbol{f} : \mathcal{D} \to \mathcal{O}$ is denoted as the feasible set of the objective space $\mathbb{R}^m$.

The solution to problem (1) is the set of trade-off points jointly minimizing the objective functions. The formalism behind the joint optimization is expressed in terms of the Pareto dominance relation. A decision vector $\boldsymbol{x}$ dominates another vector $\boldsymbol{x}'$, iff $\forall i \in \{1, \ldots, m\} : f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}')$ and $\exists i \in \{1, \ldots, m\}$:

[1] Maximization problems max $\boldsymbol{f}(\boldsymbol{x})$ can be written as min $-\boldsymbol{f}(\boldsymbol{x})$.

$f_i(\boldsymbol{x}) \neq f_i(\boldsymbol{x}')$. The subset of $\mathcal{D}$ which contains the elements that are not dominated by any other element of $\mathcal{D}$ is denoted as the Pareto-optimal set $\mathcal{D}^*$. Its image in the objective space $\mathcal{O}$ is called the Pareto-optimal front $\mathcal{O}^*$. For continuous problems, $\mathcal{D}^*$ and $\mathcal{O}^*$ usually contain an infinite number of solutions.

MOEAs are population-based stochastic optimization algorithms. Each individual in the population $\mathcal{P}$ of the MOEA represents a candidate solution. During the optimization, the individuals are improved by means of evolutionary operators, such as mutation and crossover. The image of the non-dominated individuals in objective space is denoted as non-dominated front $\mathcal{PF}^*$. The objective vectors obtained by the individuals in $\mathcal{PF}^*$ provide a finite-size approximation of $\mathcal{O}^*$. In order to evaluate the quality of the approximation set $\mathcal{PF}_t^*$ of generation t, set performance indicators have become established [21]. The target of an OSC is to detect the generation t which provides the best possible trade-off between the approximation quality of $\mathcal{PF}_t^*$ and the required generations t.

3 Taxonomy

A brief summary of theoretical single- and multi-objective convergence detection approaches based on formal conditions has already been published [16]. In this summary, also the differences between multi- and single-objective problems is discussed. If the application of formal convergence conditions is not possible, heuristic OSC are used to detect that further improvements are unlikely, or are expected to be too small – even if no formal convergence is obtained. In this paper, we are focusing on these heuristic OSC. Thus, a formal notation of convergence of a set of points in the multi-objective context is not required. The procedure of these OSC can be separated in at least two steps:

1. The expected improvement of the MOEA is evaluated.
2. Based on this improvement and a predefined threshold, a decision about stopping the MOEA is made.

For the first step, several PIs have been proposed. A straightforward approach is the use of unary performance indicators, such as convergence metric (CM) and diversity metric (DVM) [4], maximum crowding distance (maxCD) [15], or the hypervolume (HV)[20,19,9] dominated by the current $\mathcal{PF}_t^*$ with respect to a reference point which is dominated by all individuals in $\mathcal{PF}_t^*$ [6]. Moreover, binary performance indicators, such as the ε- (Epsilon) [21], and the R2-indicator (R) [10], can be used to evaluate the improvement between different MOEA generations [20,9]. In these cases, but also for some of the unary indicators (CM and DVM), a reference set is required. Since the best set available is usually the one of the current generation $\mathcal{PF}_t^*$, a recomputation of previous PI values based on this reference set can become necessary. A clear advantage of using established performance metrics consists in the availability of formal results from the theory of performance assessment [21] which can be transferred to the PI.

Nevertheless, also specialized PIs for convergence detection have been presented. Martí et al. [11,12] proposed the mutual domination rate (MDR). MDR contrasts how many individuals of $\mathcal{PF}_t^*$ dominate individuals of $\mathcal{PF}_{t-1}^*$ and vice versa. It is capable of measuring the progress of the optimization with almost no additional computational cost as it can be embedded in Pareto-based MOEAs and reuses their computations. Therefore it is suitable for solving large-scale or many-objective problems with large population sizes. If MDR equals 1 then the entire population of the iteration is better than its predecessor. For MDR $= 0$, no substantial progress has been achieved. MDR < 0 indicates a deterioration of the population. Bui et al. [3] introduced the dominance-based quality (DQP) of a set $\mathcal{PF}_t^*$. For each solution in $\mathcal{PF}_t^*$, the ratio of dominating individuals in the neighborhood of this solution are computed. The DQP is then defined as the average ratio over all solutions in $\mathcal{PF}_t^*$. DQP $= 0$ indicates that no improving solutions can be found in the neighborhood of the current solutions in $\mathcal{PF}_t^*$. For the estimation of the ratios, Monte Carlo sampling is performed around each solution in $\mathcal{PF}_t^*$. Thus, the DQP is only suitable if many additional evaluations of the objective function can be performed. Goel and Stander [8] proposed the consolidation ratio (CR). The CR is a dominance-based convergence metric based on an external archive of all non-dominated solutions found during the run of the MOEA. It is defined as the relative amount of the archive members in generation $t - t_{\text{mem}}$ which are still contained in the archive of the current generation t. In improving phases CR should be low whereas it asymptotically approaches one when convergence is achieved. This PI can be inefficient because the archive can become very large, in particular for many-objective problems.

Because of the non-deterministic nature of EAs, it can be of avail to have an *evidence gathering process* (EGP) that combines different PI values. This EGP can take into account the measurements of previous generations or more than one PI in order to increase the robustness of the approach. Different EGP approaches are discussed in the following while the descriptive notation in brackets is later on used in the formal framework in section 3.1. Many approaches [3,8] directly use the value of the PI computed in the current generation t for deciding if the MOEA is stopped (Direct). However, a single PI evaluation usually cannot provide enough information for a robust conclusion. A straightforward idea of aggregating different PI values is the use of descriptive statistics. In particular, the second moment, i. e., the standard deviation (STD) of the values, is used in order to evaluate the variability within the PI [15,20,19]. Martí et al. [11,12] propose the use of a simplified Kalman filter (Kalman). Due to the recursive formulation, the estimation at iteration t is based on all PI values gathered until then. Moreover, it considers the associated covariance error, i. e., the minimum possible error under linear conditions. A similar, but simpler, idea is proposed by Goel and Stander [8] which use a moving average as EGP (Moving). In other approaches, a linear regression analysis on the PI values of the last t_{mem} generations is performed [20,9] (Reg) in order to estimate the expected improvement and to filter out the stochastic noise in the PI evaluations.

Based on the outcome of the EGP, it can be decided whether the MOEA is stopped. Most known approaches [15,3,8] use a threshold with which the outcome is compared (Threshold). The MOEA is stopped in case the current value of the

EGP exceeds or falls below this threshold. The approaches of Martí et al. [11,12] also use confidence information (CI) based on the assumption of a normally distributed error (CInormal). The MOEA is only stopped when the estimated EGP value is below the threshold with a given probability limit. Guerrero et al. [9] do not use CIs, but ensure the quality of the regression analysis by comparing the mean squared error of the fit to a precomputed threshold (validThreshold). As an extension of this approach, Trautmann and Wagner [20,19] perform statistical tests on the outcome of the EGP. These tests are adapted to the corresponding EGP, i. e., the χ^2-test especially suited for variances (squared STD) is used with STD while a t-test is used in cases where an estimated EGP value with a normal distributed error is provided by the EGP, such as for Reg or Kalman (adaptTest). In these approaches, the p-values obtained in the tests are compared to a fixed significance level $\alpha = 0.05$. In order to further increase the robustness of the stopping decision, Wagner et al. [20] propose to wait for a repeated occurrence of the stopping condition, denoted as hits h. Moreover, the use of multiple EGPs can assist in analyzing different aspects of the PI, such as the variation (STD) and the significance of the linear trend (Reg) [20].

3.1 Formal Framework

An online stopping criterion can be formally defined as a 4-tuple,

$$\begin{aligned} OSC &:= \{\mathcal{S}, \Pi(\cdot), \Upsilon(\cdot), \Phi(\cdot)\} && \text{with} \qquad (2)\\ \mathcal{S} &: \text{ data structure,} && \text{(state of the OSC)}\\ \Pi &: \mathcal{PF}_t^* \times \mathcal{S} \to \mathcal{S}, && \text{(progress indicator (PI) computation)}\\ \Upsilon &: \mathcal{S} \to \mathcal{S}, && \text{(evidence gathering process, EGP)}\\ \Phi &: \mathcal{S} \to \{\text{true}, \text{false}\}\,. && \text{(stopping decision)} \end{aligned}$$

In the state $\mathcal{S}$, all information required for the computations of the EGP are stored. It necessarily includes the input data $\mathcal{M}$ for the EGP. In the following, we use $\mathcal{S}.\mathcal{M}$ in order to address the current version of $\mathcal{M}$ stored in the state $\mathcal{S}$. The state $\mathcal{S}$ can additionally contain previous Pareto front approximations or PI values, an external archive, or flags indicating whether the threshold has been reached in the last generations. These information can be changed or used in different functions of the taxonomy and are therefore exchanged via $\mathcal{S}$. The data stored in the state ensures that the OSC can make the stopping decision just based on the Pareto front approximation $\mathcal{PF}_t^*$ of the current generation.

The function $\Pi : \mathcal{PF}_t^* \times \mathcal{S} \to \mathcal{S}$ uses the PIs to update the input data $\mathcal{S}.\mathcal{M}$ for the EGP. This general type of function is introduced since the update can differ depending on the considered PIs, e. g., some approaches update the PI of all preceding generations based on the current generation $\mathcal{PF}_t^*$, whereas others only update the values of the last generation. Consequently, the size of $\mathcal{S}.\mathcal{M}$ can be up to $P \times t_{\text{mem}}$, where P is the number of PIs and t_{mem} is the number of preceding generations considered in the EGP. In Π also all state updates required for the PI computation, such as the update of the archive and the storage of previously computed PI values, are performed. Consequently, the input data $\mathcal{S}.\mathcal{M}$ is a necessary part of the updated state, as it would restrict the generality of the framework as sole output of Π.

The function $\Upsilon : \mathcal{S} \rightarrow \mathcal{S}$ encodes the EGP. It updates the state of the criterion based on the input data $\mathcal{S}.\mathcal{M}$ included in the current state. Usually, the EGP returns one aggregated value per PI, but also a combined analysis like in OCD [20] can be performed. In this case, the EGP value of the combined analysis is assigned to all considered PI.

The decision function $\Phi : \mathcal{S} \rightarrow \{\text{true}, \text{false}\}$ finally determines whether the current state of the criterion indicates that the expected improvement of the MOEA is below the predefined threshold ε, i. e., the MOEA should be stopped. For this decision, the EGP value, but also additional information, such as the estimation error and the degrees of freedom in the estimation of the EGP value, are usually utilized. The decision function can only return a single Boolean. If multiple EGPs are considered in parallel, also the aggregation of the corresponding decisions has to be performed in Φ.

Using these formalisms, the procedure of a generic OSC can be implemented as shown in Algorithm 1. The user has to specify the MOEA, the problem of interest and the maximum affordable number of generations $t_{\max}$, as well as PI-related data of the problem, such as a reference set and the ideal and nadir points [6]. The actual OSC is specified by the combination of the PIs, the EGPs, and the stopping decisions. For each step, also multiple functions can be provided.

After the initialization of the state in which the archive is initialized and information about the chosen PI and EGP are stored, the control parameters of the OSC are initialized. After each generation of the MOEA, $\mathcal{S}.\mathcal{M}$ and the required data structures are updated using the chosen Π_i. If there are t_{mem} measurements, the functions $\Upsilon_j(\cdot)$ are applied in order to attach the EGP value for each PI to $\mathcal{S}$. Finally, $\Phi_k(\cdot)$ can be applied to determine whether the algorithm should be stopped.

3.2 Integration of the State of the Art

In this subsection, we will present a survey of the state-of-the-art OSC in chronological publication date order. These approaches are described using the proposed formalization. A summary is provided in Table 1.

Deb and Jain: Running Metrics. Deb and Jain [4] were the first authors who proposed the investigation of performance metrics over the run of the MOEA. They used two metrics, one for evaluating the convergence and one for measuring the diversity of $\mathcal{PF}_t^*$. The convergence metric (CM) calculates the average of the smallest normalized euclidean distance from each individual in $\mathcal{PF}_t^*$ to a precomputed reference set. For the computation of the diversity metric (DVM), all objective vectors of $\mathcal{PF}_t^*$ are projected onto a hyperplane of dimension $m-1$ which is then uniformly divided into discrete grid cells. The DVM tracks the number of attained grid cells and also evaluates the distribution by assigning different scores for predefined neighborhood patterns. In order to avoid bad DVM values based on unattainable grid cells, again a reference set is used. The

Algorithm 1. Implementation of an OSC using the taxonomy definition (eq. 2)

```
General parameters:
  • Multi-objective evolutionary algorithm of interest.
  • Multi-objective problem of interest.
  • $t_{\max}$, maximum number of iterations.
  • $\mathcal{PI}$, set of PI functions $\Pi_i$.
  • $\mathcal{EGP}$, set of EGP functions $\Upsilon_j$.
  • $\mathcal{SDF}$, set of stopping decision functions $\Phi_k$, $k = \{1, \ldots, K\}$.
  • Problem-based parameters (reference set, ideal and nadir points).
  • Manually defined settings of control parameters (optional).
Initialize state $\mathcal{S}$.
Initialize control parameters of $\Pi_i$, $\Upsilon_j$, and $\Phi_k$.
$t = 0$.
while $t < t_{\max}$ do
  $t = t + 1$.
  Perform one generation of the MOEA and obtain $\mathcal{PF}^*_t$.
  for each indicator $\Pi_i$ in $\mathcal{PI}$ do
    Update input data $\mathcal{S}.\mathcal{M}$ and PI-dependent information, $\mathcal{S} = \Pi_i(\mathcal{PF}^*_t, \mathcal{S})$.
  end for
  if $|\mathcal{S}.\mathcal{M}| = t_{\text{mem}}$ then
    for each EGP $\Upsilon_j$ in $\mathcal{EGP}$ do
      Update EGP value based on $\mathcal{S}.\mathcal{M}$, $\mathcal{S} = \Upsilon_j(\mathcal{S})$.
    end for
    for each stopping decision function $\Phi_k$ in $\mathcal{SDF}$ do
      Compute stop decision, stop(k) = $\Phi_k(\mathcal{S})$
    end for
    if $\forall k$ : stop(k) = true then
      Stop MOEA!
      return  $t$ and $\mathcal{S}$.
    end if
  end if
end while
```

EGP and the final decision then rely on a visual inspection of the progression of the CM and DVM by the user. Consequently, the state $\mathcal{S}$ of this criterion contains the reference set and all values of the CM and DVM computed until the current generation.

Rudenko and Schoenauer: Stability Measure. Rudenko and Schoenauer [15] defined a stability measure for the $\mathcal{PF}^*_t$ of NSGA-II [5]. Their experimental studies showed that the stagnation of the maximum crowding distance (maxCD) within $\mathcal{PF}^*_t$ is a suitable indicator for NSGA-II convergence. Thus, the standard deviation of the last t_{mem} values of the maximum crowding distance is used as EGP (STD). For the computation, the last $t_{\text{mem}} - 1$ values of $maxCD$ are contained in the state $\mathcal{S}$. In each generation, $\mathcal{S}$ is updated using the current $maxCD$ value and STD is computed. The decision step requires a user defined threshold ε leading to an NSGA-II termination once the STD falls below this value (Threshold).

Table 1. Definition of the taxonomy functions in known approaches. The acronoyms and abbreviations are introduced in section 3.

Approach	$\Pi(\mathcal{PF}^*_t, \mathcal{S})$	$\mathcal{S}$	$\Upsilon(\mathcal{S})$	$\Phi(\mathcal{S})$	Parameters (default)
Running metrics [4]	CM, DVM $\lvert\mathcal{S}.\mathcal{M}\rvert = 2 \times 1$	- $\mathcal{M}$: All CM and DVM values - Reference set	Attach current CM and DVM values to state	Visual check by decision maker	none
Stability measure [15]	maxCD $\lvert\mathcal{S}.\mathcal{M}\rvert = 1 \times 1$	- $\mathcal{M}$: maxCD of generation $(t - t_{\text{mem}})$ to t - STD	STD of $\mathcal{S}.\mathcal{M}$	Threshold	- t_{mem} (40) - ε (0.02) - hits h (1)
MBGM [11, 12]	MDR $\lvert\mathcal{S}.\mathcal{M}\rvert = 1 \times 1$	- $\mathcal{M}$: Current MDR value - $\mathcal{PF}^*_{t-1}$ - Kalman state with corresp. STD	Kalman	CInormal	- ε (0) - p-CI (97.725 %) - Kalman inertia R (0.1) - hits h (1)
OCD-Classic [20, 14, 19]	Eps, R, HV (parallel) $\lvert\mathcal{S}.\mathcal{M}\rvert = 3 \times t_{\text{mem}}$	- $\mathcal{M}$: Eps, R, and HV of generation $(t - t_{\text{mem}})$ to $(t-1)$ - $\mathcal{PF}^*$ of generation $(t - t_{\text{mem}})$ to $(t-1)$ - Current slope β with corresp. STD - Current STD - p-values of generations t and $(t-1)$	- STD of each PI in $\mathcal{S}.\mathcal{M}$ - Reg on $\mathcal{S}.\mathcal{M}$ (individually standardized)	- χ^2-test: STD $< \varepsilon$ - t-test: $\beta = 0$	- t_{mem} (16) - ε (0.001) - hits h (2)
DQP [3]	DQP $\lvert\mathcal{S}.\mathcal{M}\rvert = 1 \times 1$	$\mathcal{M}$: Current DQP value	Direct	Visual check by decision maker	- samples N (500) - radius r (0.05)
LSSC [9]	Eps, HV, MDR (separately) $\lvert\mathcal{S}.\mathcal{M}\rvert = 1 \times t_{\text{mem}}$	- $\mathcal{M}$: Eps, HV, or MDR of generation $(t - t_{\text{mem}})$ to $(t-1)$ - $\mathcal{PF}^*$ of generation $(t - t_{\text{mem}})$ to $(t-1)$ - Current slope β	Reg on each PI in $\mathcal{S}.\mathcal{M}$	validThreshold	- t_{mem} (30) - ε (HV: 0.002, Epsilon: 0.0004, MDR: 0.00002) - hits h (1)
OCD-HV [19]	HV $\lvert\mathcal{S}.\mathcal{M}\rvert = 1 \times t_{\text{mem}}$	- $\mathcal{M}$: Differences between the HVs of generation $(t - t_{\text{mem}})$ to $(t-1)$ and t - p-values of generations t and $(t-1)$	STD of $\mathcal{S}.\mathcal{M}$	- χ^2-test: STD $< \varepsilon$	- t_{mem} (14) - ε (0.0001) - hits h (2)
CR [8]	CR $\lvert\mathcal{S}.\mathcal{M}\rvert = 1 \times 1$	- $\mathcal{M}$: CR of generation $(t - t_{\text{mem}})$ to t - Archive of non-dominated ind. - $\varepsilon_{adaptive}$ - $\text{U*}_{t-t_{mem}}$	Direct or Moving average	CR $> \varepsilon$ or $\text{U*}_t < \varepsilon_{adaptive}$	- t_{mem} (10) - ε (0.8) - utility ratio F (10) - hits h (1)

Martí et al.: MGBM Criterion. Martí, García, Berlanga, and Molina [11,12] proposed the MGBM criterion (according to the authors' last names), which combines the mutual domination rate (MDR) with a simplified Kalman filter that is used as EGP. The function $\mathit{\Pi}$ considers $\mathcal{PF}_{t-1}^*$ and $\mathcal{PF}_t^*$ and applies the MDR indicator to update $\mathcal{S}.\mathcal{M}$. Thus, the Pareto front of the previous generation has to be stored in the state $\mathcal{S}$. The EGP function $\mathit{\Upsilon}$ applies the Kalman filter and updates the Kalman state and the corresponding estimated error in $\mathcal{S}$. The decision function $\mathit{\Phi}$ is realized by stopping the MOEA when the confidence interval of the a-posteriori estimation completely falls below the prespecified threshold ε.

Wagner et al.: Online Convergence Detection (OCD). In the Online Convergence Detection [20] approach, the established performance measures HV, R2- and additive ε-indicator are used as PIs. The function $\mathit{\Pi}$ updates all t_{mem} PI values stored in $\mathcal{S}.\mathcal{M}$ using the current generation $\mathcal{PF}_t^*$ as reference set. Consequently, the sets $\mathcal{PF}_{t-t_{\mathrm{mem}}}^*$ to $\mathcal{PF}_{t-1}^*$ have to be additionally stored in the state $\mathcal{S}$. In $\mathit{\Upsilon}$, the variance of the values in $\mathcal{S}.\mathcal{M}$ is computed for each PI. Moreover, a least-squares fit of a linear model with slope parameter β is performed based on the individually standardized values in $\mathcal{S}.\mathcal{M}$. In $\mathit{\Phi}$, the variance is then compared to a threshold variance ε by means of the one-sided χ^2-variance test with H_0: $\mathrm{VAR}(\mathcal{S}.\mathcal{M}) \geq \varepsilon$ and a p-value is looked up. By testing the hypothesis H_0: $\beta = 0$ by means of a t-test, a second p-value is obtained. For these tests, the variance obtained by STD, β, and its standard error have to be stored in the state. The same holds for the resulting p-values. The MOEA is stopped when the p-values of two consecutive generations are below the critical level $\alpha = 0.05$ for one of the variance tests (the null hypothesis H_0 is rejected) or above $\alpha = 0.05$ for the regression test (H_0 is accepted). Consequently, the p-values of the preceding generations have to be stored in $\mathcal{S}$.

In [19] a reduced variant of the OCD approach for indicator-based MOEA was introduced. This approach was illustrated for the HV indicator and the SMS-EMOA [1] (OCD-HV). Since the HV is a unary indicator, only the absolute HV values have to be stored. The previous $\mathcal{PF}_t^*$ can be neglected. For better compliance with the other PI, the differences to the value of the current set $\mathcal{PF}_t^*$ are stored in $\mathcal{S}.\mathcal{M}$ in order to minimize the PI. In case the internally optimized performance indicator monotonically increases, as for the SMS-EMOA and the HV, OCD should only consider this PI. The regression test can be neglected. Consequently, the complexity of OCD is reduced by concentrating on the variance test for one specific PI.

Bui et al.: Dominance-Based Quality of $\mathcal{P}$ (DQP). Bui et al. [3] introduce a dominance-based stability measure which approximately evaluates the local optimality of a solution (DQP). The DQP is the only PI that requires many additional evaluations of the objective function for estimating the ratio of dominating solutions in the neighborhood of a solution. A Monte Carlo simulation with 500 evaluations per solution in $\mathcal{PF}_t^*$ was used. Consequently, the DQP is a very expensive, but powerful measure. No additional state informations or EGPs are required. No clear guidelines for stopping the MOEA are provided. Instead, a visual analysis of the convergence behavior and possible stagnation phases is performed. However, a clear stopping criterion would be $DQP = 0$, as this would be the case when no local improvements are possible. In fact DQP is closely related to the gradient of a solution in single-objective optimization. In line with this observation, the authors also use DQP as measure for guiding a local search [3].

Guerrero et al.: Least Squares Stopping Criterion (LSSC). LSSC [9] can be seen as an approach to integrate both EGP of OCD into a single EGP and to also simplify the PI computation and the stopping decision. Therefore, only one PI is considered and the variance-based EGP and the statistical tests for the stopping decision are omitted. Still, a regression analysis of the PI is performed as EGP and the PI values of the last t_{mem} generations are updated using the current generation as reference set. Thus, the last t_{mem} Pareto front approximations have to be stored in the state $\mathcal{S}$ in order to update $\mathcal{S}.\mathcal{M}$. In contrast, the PIs are not standardized allowing the estimation of the expected improvement by means of the slope β. If β falls below the predefined threshold ε, the MOEA is stopped. In order to prevent a loss of robustness by omitting the statistical tests, a threshold for a goodness-of-fit test based on the regression residuals is computed via the Chebyshev inequality. Only if the model is valid, the estimated slope is compared to ε. Consequently, the analyses performed in OCD and LSSC differ. Whereas LSSC directly tries to detect whether the expected improvement falls below the allowed threshold ε, OCD tests the significance of the linear trend whereas the magnitude of the expected improvement is evaluated via the variance of $\mathcal{S}.\mathcal{M}$.

Goel and Stander: Non-dominance based convergence metric. Goel and Stander [8] use a dominance-based PI based on an external archive of non-dominated solutions which is updated in each generation. The current archive is stored in $\mathcal{S}$ and is used to determine the CR. The authors provide empirical evidence for the robustness of the CR, so that no EGP is applied (Direct). The stopping decision is made by comparing the CR with a predefined threshold ε (Threshold).

In addition, an utility-based approach is proposed. The utility is defined as the difference in the CR between the generations t and $t - t_{\text{mem}}$. In order to increase the robustness of the approach, a moving average $U_t^* = (U_t + U_{t-t_{\text{mem}}})/2$ is used as EGP (Moving). The MOEA is stopped when the utility falls below an adaptively computed threshold $\varepsilon_{\text{adaptive}}$. Moreover, a minimum CR of $\text{CR}_{\text{min}} = 0.5$ has to be reached in order to avoid a premature stopping due to perturbances in early generations. The adaptive threshold $\varepsilon_{\text{adaptive}}$ is defined as the fraction $\text{CR}_{\text{init}}/(F \cdot t_{\text{init}})$ of the initial utility U_{init}, which corresponds to the first CR value CR_{init} exceeding 0.5 and the corresponding generation t_{init}. F is a user parameter that specifies which ratio of the averaged initial utility $\text{CR}_{\text{init}}/t_{\text{init}}$ is at least acceptable. For this version, also $\varepsilon_{\text{adaptive}}$ and $\text{U*}_{t-t_{\text{mem}}}$ have to be stored in $\mathcal{S}$.

3.3 Discussion

Basically, the existing PIs can be classified with respect to their optimization goal. One class is formed by the PIs based on analyzing the dominance relation between the current population (or archive) and a previous one, e. g., MDR and CR. Other approaches provide information about the distribution (maxCD, DVM) or local optimality of the solutions (DQP). Only a few of the PI try to combine some of these goals, e. g., HV, R, Epsilon, and CM, each with different trade-offs.

The dominance-based PI the convergence of the population to be formally assessed. The probability of improving the diversity and distribution and therewith the quality of the discrete approximation of O^* is not specifically addressed. The improvements in these PI will therefore reduce much faster. Moreover, the magnitude of the improvement generated by a new non-dominated solution is not considered. This information would be important in order to evaluate an expected improvement. As shown in the last years [17], the dominance relation has only a weak explanatory power for many-objective problems.

The dominance-based PI usually reuse the information provided by the selection method of the MOEA. Thus, they do not require expensive additional calculations. PIs like CM, R, and HV have to be additionally computed in each MOEA generation, where especially the dominated hypervolume has a complexity which increases exponentially with the objective space dimension. Bui et al. [3] even perform additional evaluations for convergence detection. In general, the use of additional computational time or evaluations should be kept below the effort of the alternative option of just allowing the MOEA to precede for an affordable number of additional generations.

In addition, reference and nadir points, as well as reference sets, can be required for some PIs, e. g., the reference set for the CM and DVM, the ideal and nadir point for R2, and the reference point for HV. In contrast to mathematical test cases, this information is usually not existing for practical applications. Strategies to obtain this data have to be derived which could comprise preliminary algorithm runs, random sampling, or evaluations on a grid covering the whole search space. Based on approximations of the objective boundaries, the normalization of the PI to unit intervals is possible – an approach that is often recommended [4,21]. However, even the normalization can lead to scalarization effects which make the specification of thresholds difficult [19]. For the dominance-based indicators, usually relative amounts are calculated, e.g., $-1 \leq MDR \leq 1$ or $0 \leq CR \leq 1$, which facilitate the definition of adequate threshold values. Nevertheless, the only reasonable threshold for these approaches is $\varepsilon = 0$ based on the above considerations.

Some methods do not use a distinct EGP. They rely on a single evaluation of the considered PI. Due to the stochastic nature of MOEAs, it is obvious that those approaches will not be as robust as alternative ones using an EGP gathering PIs over a time window. Moreover, the EGP-based approaches are usually flexible with respect to the kind of integrated PI. By means of a suitable PI, the performance aspects (e. g., convergence, distribution, spread) which are the most important for the optimization task at hand can be considered in the OSC. In this context, also the considered MOEA has an important role. Mathematical convergence can only be expected if the corresponding MOEA is based on this PI, e. g., the SMS-EMOA in combination with the HV [2]. Furthermore, most OSC are designed for separately using a single PI. As performance of a MOEA has different aspects [4,21], it should be analyzed if the usage of PIs covering these aspects of the approximation quality could support an efficient OSC decision.

Another important OSC design issue is concerned with the choice of the stopping decision. Statistical tests or confidence intervals lend themselves to draw robust decisions from random variables monitored over time. However, in order to choose an adequate test or distribution, some assumptions on the behavior of the considered PI are necessary. As a first approach, Mersmann et al. [13] analyze the distribution of the final HV value of different MOEAs. Among other characteristics it is shown to be unimodal in most cases. Consequently, the use of classical tests is possible, maybe based on additional transformations.

The parametrization of the OSC requires special attention as well. Parameters have to be carefully chosen in order to obtain the desired results with respect to the trade-off between runtime and approximation quality. For most approaches, no clear guidelines for setting up the required parameters are given or a visual analysis is suggested [4,3]. In contrast, Wagner and Trautmann [19] empirically derive guidelines for reasonably setting the OCD parameters t_{mem} and ε based on statistical design-of-experiment methods. The resulting parameter recommendations can be found in Table 1. For reasonable comparisons between the OSC, such kind of studies should also be performed for the other OSC. Furthermore, the problems and possibilities resulting from a combination of the methods with respect to the proposed PI, EGP, and stopping decisions should be a matter of future research. In this context, an analysis of the compatibility of the PI, EGP, and decision criteria would be of special interest.

4 MATLAB Toolbox for Online Stopping Criteria

In the previous subsection, many open questions in the field of OSC are discussed. However, all choices of test problems and MOEAs for the analysis of OSC put a subjective bias to the results. In order to assist researchers in analyzing these questions, a MATLAB toolbox based on the OSC taxonomy was implemented [18]. Thus, the framework allows the application of the OSC to the test problems and favorite MOEAs of the user. Based on the framework, an interested user can analyze and tune the OSC on his specific setup.

The framework follows the pseudocode provided in Algorithm 1. It allows the arbitrary combination of building blocks which can be used to design an adapted OSC for the specific task at hand. Consequently, the analysis of the compatibility of different subfunctions can directly be performed. Within the framework, the abbreviations of section 3 are used to address the corresponding subfunctions. Accordingly, each subfunction is documented in this paper.

The control parameters of the OSC are initialized automatically using default values. This procedure prevents the user from searching for default values of the parameters before using the framework and also encourages researchers to perform parameter studies before proposing an OSC. Nevertheless, experienced users have the opportunity to specify some of the parameters on their own using a options structure.

In order to allow arbitrary combinations of Π, Υ, and Φ, as proposed in Algorithm 1, some additional features are integrated within the MATLAB framework:

- The use of different stopping decisions in parallel is possible.
- It is possible to combine the stopping decisions for different PI and EGP by more than the already proposed rule: all PI for at least one EGP [20]. Further possibilities are: all, any, all EGP for at least one PI, and a majority voting.
- The standard deviation of STD-EGP is calculated using bootstrapping [7].

The choice of allowing multiple stopping decisions in parallel is motivated by the different amounts of information provided by the different EGP. The CI- and t-test-based approaches require EGP that also provide error estimates. By combining these methods with the threshold decision, which will always stop when the CI- or test-based approaches would, also these EGP can be handled. Thus, if some information is missing, e. g., the standard deviation after applying the Direct EGP, the adaptTest- or CInormal-EGP are ignored and only Threshold decision is used. This enhancement makes the framework more flexible for new conceptually different stopping decisions.

By means of the formalization through the taxonomy, the interfaces for the framework are clearly defined. Researchers in the field of OSC can easily integrate their methods by structuring their OSC following the taxonomy. Then each subfunction is implemented within the framework, and a benchmark with all state-of-the-art OSC can directly be performed. As a side effect, a systematic integration of new OSC into the state of the art is promoted.

5 Conclusion and Outlook

In this paper, a comprehensive overview of sophisticated heuristic online stopping criteria (OSC) for EA-based multi-objective optimization is provided. The approaches are integrated into a taxonomy by splitting them into their building blocks which cover the different steps to be performed when applying an OSC. The presented taxonomy allows comparisons of OSC approaches to be systematically performed. The analysis of the strengths and weaknesses of a specific OSC can be broken down to the responsible subfunctions, e. g., the methods can be classified by the kind of PI used, the complexity of the EGP, and the integration of statistical techniques in the decision making. Concluding, OSC methods relying on an EGP with respect to PIs gathered from preceding generations are likely to be more robust, but computationally expensive. The additional use of statistical techniques can further increase robustness, but needs to be adapted to the data of the EGP. In contrast, complex and expensive PI like DQP may not require a sophisticated EGP or stopping decision.

The parametrization of an individual OSC is not an easy task and strongly influences its performance. Unfortunately, sufficient and comprehensive guidelines for the required parameter settings are only presented for a small subset of the OSC strategies. Moreover, the recommended thresholds for the specific PI are different, making a fair comparison almost impossible. In order to simplify a systematic comparison, a MATLAB toolbox [18] was implemented. This toolbox is structured according to the building blocks of the presented taxonomy and all approaches discussed in this paper were integrated. By means of this toolbox, the expert can evaluate the approaches – and also combinations of them – on his problem and can then choose the OSC which provides the best performance with regard to his objectives.

A systematic evaluation and comparison of all presented approaches will be the main focus of our future research. This includes a parameter tuning, as well as the combination of algorithmic concepts. To accomplish this, a systematic performance assessment of OSC has to be proposed and discussed.

Acknowledgments

This paper is based on investigations of the collaborative research centers SFB/TR TRR 30 and SFB 823, which are kindly supported by the Deutsche Forschungsgemeinschaft (DFG). L. Martí acknowledges support from projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM CONTEXTS S2009/TIC-1485 and DPS2008-07029-C02-02.

References

1. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)
2. Beume, N., Laumanns, M., Rudolph, G.: Convergence rates of (1+1) evolutionary multiobjective optimization algorithms. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 597–606. Springer, Heidelberg (2010)
3. Bui, L.T., Wesolkowski, S., Bender, A., Abbass, H.A., Barlow, M.: A dominance-based stability measure for multi-objective evolutionary algorithms. In: Tyrrell, A., et al. (eds.) Proc. Int'l. Congress on Evolutionary Computation (CEC 2009), pp. 749–756. IEEE Press, Piscataway (2009)
4. Deb, K., Jain, S.: Running performance metrics for evolutionary multi-objective optimization. In: Simulated Evolution and Learning (SEAL), pp. 13–20 (2002)
5. Deb, K., Pratap, A., Agarwal, S.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6(8) (2002)
6. Deb, K., Miettinen, K., Chaudhuri, S.: Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. Trans. Evol. Comp. 14, 821–841 (2010)

7. Efron, B.: Bootstrap methods: Another look at the jackknife. Annals of Statistics 7(1), 1–26 (1979)
8. Goel, T., Stander, N.: A non-dominance-based online stopping criterion for multi–objective evolutionary algorithms. International Journal for Numerical Methods in Engineering (Online access) (2010) doi: 10.1002/nme.2909
9. Guerrero, J.L., Martí, L., García, J., Berlanga, A., Molina, J.M.: Introducing a robust and efficient stopping criterion for MOEAs. In: Fogel, G., Ishibuchi, H. (eds.) Proc. Int'l. Congress on Evolutionary Computation (CEC 2010), pp. 1–8. IEEE Press, Piscataway (2010)
10. Hansen, M.P., Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Tech. Rep. IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark (1998)
11. Martí, L., García, J., Berlanga, A., Molina, J.M.: A cumulative evidential stopping criterion for multiobjective optimization evolutionary algorithms. In: Thierens, D., et al. (eds.) Proc. of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO 2007), p. 911. ACM Press, New York (2007)
12. Martí, L., García, J., Berlanga, A., Molina, J.M.: An approach to stopping criteria for multi-objective optimization evolutionary algorithms: The MGBM criterion. In: Tyrrell, A., et al. (eds.) Proc. Int'l. Congress on Evolutionary Computation (CEC 2009), pp. 1263–1270. IEEE Press, Piscataway (2009)
13. Mersmann, O., Trautmann, H., Naujoks, B., Weihs, C.: On the distribution of EMOA hypervolumes. In: Blum, C., Battiti, R. (eds.) LION 4. LNCS, vol. 6073, pp. 333–337. Springer, Heidelberg (2010)
14. Naujoks, B., Trautmann, H.: Online convergence detection for multiobjective aerodynamic applications. In: Tyrrell, A., et al. (eds.) Proc. Int'l. Congress on Evolutionary Computation (CEC 2009), pp. 332–339. IEEE press, Piscataway (2009)
15. Rudenko, O., Schoenauer, M.: A steady performance stopping criterion for pareto-based evolutionary algorithms. In: The 6th International Multi-Objective Programming and Goal Programming Conference, Hammamet, Tunisia (2004)
16. Trautmann, H., Wagner, T., Preuss, M., Mehnen, J.: Statistical methods for convergence detection of multiobjective evolutionary algorithms. Evolutionary Computation Journal, Special Issue: Twelve Years of EC Research in Dortmund 17(4), 493–509 (2009)
17. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., et al. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007)
18. Wagner, T., Martí, L.: Taxonomy-based matlab framework for online stopping criteria (2010), http://www.giaa.inf.uc3m.es/miembros/lmarti/stopping
19. Wagner, T., Trautmann, H.: Online convergence detection for evolutionary multi-objective algorithms revisited. In: Fogel, G., Ishibuchi, H. (eds.) Proc. Int'l. Congress on Evolutionary Computation (CEC 2010), pp. 3554–3561. IEEE press, Piscataway (2010)
20. Wagner, T., Trautmann, H., Naujoks, B.: OCD: Online convergence detection for evolutionary multi-objective algorithms based on statistical testing. In: Ehrgott, M., et al. (eds.) EMO 2009. LNCS, vol. 5467, pp. 198–215. Springer, Heidelberg (2009)
21. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 8(2), 117–132 (2003)

Not All Parents Are Equal for MO-CMA-ES*

Ilya Loshchilov[1,2], Marc Schoenauer[1,2], and Michèle Sebag[2,1]

[1] TAO Project-team, INRIA Saclay - Île-de-France
[2] Laboratoire de Recherche en Informatique (UMR CNRS 8623)
Université Paris-Sud, 91128 Orsay Cedex, France
FirstName.LastName@inria.fr

Abstract. The Steady State variants of the Multi-Objective Covariance Matrix Adaptation Evolution Strategy (SS-MO-CMA-ES) generate one offspring from a uniformly selected parent. Some other parental selection operators for SS-MO-CMA-ES are investigated in this paper. These operators involve the definition of multi-objective rewards, estimating the expectation of the offspring survival and its Hypervolume contribution. Two selection modes, respectively using tournament, and inspired from the Multi-Armed Bandit framework, are used on top of these rewards. Extensive experimental validation comparatively demonstrates the merits of these new selection operators on unimodal MO problems.

1 Introduction

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [9,8] is considered today as the state-of-the art method for continuous optimization at large, at least for small to medium-sized search space (up to dimension 100) [7]. Its efficiency mostly derives from its invariance properties; not only is it invariant with respect to monotonous transformations of the objective function, like all comparison-based optimization algorithms; it is also invariant with respect to orthogonal transformations of the coordinate system, thanks to the on-line adaptation of the covariance matrix of the Gaussian mutation. The multi-objective version of CMA-ES proposed by Igel et al. [10], called MO-CMA-ES, benefits from these invariance properties (though the hypervolume indicator is not invariant) and performs very well on non-separable problems like the IHR family. MO-CMA-ES proceeds as a $(\mu + \mu)$ algorithm, where μ parents give birth to μ offspring, and the best μ individuals in the sense of Pareto dominance (out of parents plus offspring) become the parents of the next generation. As shown by [4] however, Evolutionary Multi-Objective Optimization can benefit from steady-state strategies. Accordingly, two steady state variants, hereafter called SS-MO-CMA-ES, have been proposed by Igel et al. [12], implementing some $(\mu + 1)$ selection strategy: a parent is selected uniformly (either from the whole parent population, or among the non-dominated parents), and is used to generate a single offspring, which is inserted back into the population at each time

* Work partially funded by FUI of System@tic Paris-Region ICT cluster through contract DGT 117 407 *Complex Systems Design Lab* (CSDL).

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 31–45, 2011.

step. Significant improvements over the generational version were reported on unimodal benchmark problems.

The present paper investigates some alternative choices of the fertile parent in SS-MO-CMA-ES, based on the conjecture that not all (non-dominated) parents are equal. Several indicators, estimating the expectation of offspring survival or its hypervolume contribution, are considered. These indicators are exploited within a simple tournament selection, or borrowing Multi-Armed Bandits principles [2] to deal with the Exploration vs Exploitation dilemma.

This paper is organized as follows. Section 2 recalls the basics of MO-CMA-ES; the generational and the steady state variants are described within a generic scheme. Section 3 details the proposed parent selection operators and how they fit in the generic scheme. These operators involve a rewarding procedure estimating the goodness of parents, and a selection procedure. In Section 4 the resulting algorithms are experimentally assessed on some well-known benchmark functions, comparatively to the previous versions of MO-CMA-ES, and the paper concludes with some perspectives for further research in Section 5.

2 State of the Art

This section briefly recalls the formal background of multi-objective optimization, and the basics of MO-CMA-ES and SS-MO-CMA-ES, referring the reader to [10] and [12] for a more comprehensive description.

Let $\mathcal{D} \subseteq \mathbb{R}^d$ be the decision space, and let $f_1, \ldots, f_m$ denote m objectives defined on the decision space ($f_i : \mathcal{D} \mapsto \mathbb{R}$). The objective space is given by $\mathbb{R}^m$ and the image of x in the objective space is defined as $o_x = (f_1(x), \ldots f_m(x))$. Given a pair of points $(x, y) \in \mathcal{D}$, it is said that x *dominates* y (denoted $x \prec y$) iff x is not worse than y over all objectives, and x is strictly better than y on at least one objective. It is said that $o_x \prec\ o_y$ iff $x \prec y$.

2.1 MO-CMA-ES

Originally, MO-CMA-ES involves a set of μ $(1+1)$-CMA-ES, each of which performs step-size and covariance matrix updates based on its own evolution path, and a Pareto-based survival selection mechanism that selects μ individuals from the population of size 2μ built from all parents and offspring.

Regarding the $(1+1)$-ES, the general rules used for the adaptation of the step-size and the covariance matrix in CMA-ES [9,8] cannot be used within the $(1+1)$ setting. Specific rules have hence been proposed [11], based on the success rate of the previous evolution steps, *à la* 1/5th rule [16]. The detailed description of those rules fall outside the scope of this paper, though their formal description is given in lines 8-16 in Algorithm 1 for the sake of reproducibility.

Regarding the survival selection mechanism, it is inspired by the Non-dominated Sorting procedure first proposed within the NSGA-II algorithm [6]. Two hierarchical criteria are used in turn: the Pareto rank, and the hypervolume contribution [4], that replaces the original crowding distance. Let $A = \{a_1, \ldots, a_p\}$ denote a set of p points of the objective space.

Pareto Ranking. The Pareto ranks w.r.t. A of the points in A are iteratively determined. All non-dominated points in A (denoted $ndom_1(A)$ or simply $ndom(A)$), are given rank 1. The set $ndom(A)$ is then removed from A; from this reduced set, the non-dominated points (denoted $ndom_2(A)$) are given rank 2; the process continues until all points of A have received a Pareto rank. The Pareto rank of point $a \in A$ is denoted $PR(a, A)$.

Hypervolume contribution. The *Hypervolume* of a set of points A is sometimes also called "S-Metric" [18]. Let a_{ref} denote a reference point, dominated by all points in A. The hypervolume of A is then the volume of the union of the hypercubes defined by one point of the set and a_{ref}. Formally,

$$H(A) = Volume(\bigcup_{i=1}^{i=p} Rect(a_i, a_{ref}))$$

where $Rect(a, b)$ is the hyper-rectangle whose diagonal is the segment $[ab]$. It is clear that only the non-dominated points in A (i.e. the points in $ndom(A)$) contribute to the hypervolume. The *Hypervolume contribution* of some non-dominated point a is defined as the difference between the hypervolume of the whole set A and that of the set from which a has been removed.

$$\Delta H(a, A) = H(A) - H(A \backslash \{a\})$$

For dominated points, the hypervolume contribution can also be defined by considering only the points that have the same rank. More precisely, if $PR(a) = k$, i.e. $a \in ndom_k(A)$, then

$$\Delta H(a, A) = H(ndom_k(A)) - H(ndom_k(A) \backslash \{a\})$$

Survival Selection in MO-CMA-ES. All above definitions are extended to points in the decision space as follows. Given a set $X = \{x_1, \dots x_p\}$ in the decision space, given the set $A = \{o_{x1}, \dots, o_{xp}\}$ of their image in the objective space, the Pareto rank (resp. hypervolume contribution) of any point x in X is set to the Pareto rank (resp. hypervolume contribution) of o_x in A.

Using Pareto ranking as first criterion, and the hypervolume contribution as secondary criterion (rather than the crowding distance proposed with the original NSGA-II, as advocated in [4]), a total preorder relation $\prec_X$ is defined on any finite subset X of the decision space, as follows:

$$\begin{aligned} x \prec_X y \Leftrightarrow PR(x, X) < PR(y, X) & \qquad \textit{// lower Pareto rank} \\ \text{or} \qquad & \textit{// same Pareto rank and higher HC} \\ PR(x, X) = PR(y, X) \text{ and } \Delta H(x, X) > \Delta H(y, X) & \end{aligned} \tag{1}$$

Ties on hypervolume contributions are broken at random.

Specific care must be taken with the extreme points of the set, i.e. the points for which the hypervolume contribution depends on the choice of the reference

Algorithm 1. $(\mu+\lambda)$-MO-CMA-ES Generic MO-CMA-ES scheme

```
1: $g \leftarrow 0$, initialize parent population $Q^{(0)}$;
2: repeat
3:   for $k = 1, \ldots, \lambda$ do
4:     $\underline{i_k \leftarrow ParentSelection(Q^{(g)}, k)}$;
5:     $a'^{(g+1)}_k \leftarrow a^{(g)}_{i_k}$;
6:     $\mathbf{x}'^{(g+1)}_k \sim \mathbf{x}^{(g)}_{i_k} + \sigma^{(g)}_{i_k} \mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}_{i_k}\right)$;
7:     $Q^{(g)} \leftarrow Q^{(g)} \cup \left\{a'^{(g+1)}_k\right\}$;
8:   for $k = 1, \ldots, \lambda$ do
9:     $\bar{p'}^{(g+1)}_{\mathtt{succ},k}, \bar{p}^{(g)}_{succ,i_k} \leftarrow (1-c_p)\bar{p'}^{(g+1)}_{\mathtt{succ},k} + c_p \mathtt{succ}_{Q^{(g)}}\left(a^{(g)}_{i_k}, a'^{(g+1)}_k\right)$;
10:    $\sigma'^{(g+1)}_k, \sigma^{(g)}_{i_k} \leftarrow \sigma'^{(g+1)}_k \mathtt{exp}\left(\frac{1}{d} \frac{\bar{p'}^{(g+1)}_{\mathtt{succ},k} - p^{\mathtt{target}}_{\mathtt{succ}}}{1 - p^{\mathtt{target}}_{\mathtt{succ}}}\right)$;
11:    if $\bar{\mathbf{p}'}^{(g+1)}_{\mathtt{succ},k} < \mathbf{p}_{\mathtt{thresh}}$ then
12:      $\mathbf{p}'^{(g+1)}_{c,k} \leftarrow (1-c_c)\mathbf{p}'^{(g+1)}_{c,k} + \sqrt{c_c(2-c_c)} \frac{\mathbf{x}'^{(g+1)}_k - \mathbf{x}^{(g)}_{i_k}}{\sigma^{(g)}_{i_k}}$;
13:      $\mathbf{C}'^{(g+1)}_k \leftarrow (1-c_{\mathsf{cov}})\mathbf{C}'^{(g+1)}_k + c_{\mathsf{cov}} \mathbf{p}'^{(g+1)}_{c,k} \mathbf{p}'^{(g+1)^T}_{c,k}$;
14:    else
15:      $\mathbf{p}'^{(g+1)}_{c,k} \leftarrow (1-c_c)\mathbf{p}'^{(g+1)}_{c,k}$;
16:      $\mathbf{C}'^{(g+1)}_k \leftarrow (1-c_{\mathsf{cov}})\mathbf{C}'^{(g+1)}_k + c_{\mathsf{cov}} \left(\mathbf{p}'^{(g+1)}_{c,k} \mathbf{p}'^{(g+1)^T}_{c,k} + c_c(2-c_c)\mathbf{C}'^{(g+1)}_k\right)$;
17:  $Q^{(g+1)} \leftarrow \left\{Q^{(g)}_{\prec:i} | 1 \le i \le \mu\right\}$;   // Deterministic Selection according to $\prec_{Q^{(g)}}$
18:  $\underline{ComputeRewards(Q^{(g)}, Q^{(g+1)})}$;
19:  $g \leftarrow g + 1$;
20: until stopping criterion is met.
```

point. By convention, they are associated an infinite hypervolume contribution, and they thus dominate in the sense of Eq. (1) all points with same Pareto rank.

The survival selection in MO-CMA-ES finally proceeds as the standard deterministic selection of the $(\mu + \mu)$-ES algorithm: at generation g, after each of the μ parents has generated one offspring, let $Q^{(g)}$ denote the union of the μ parents and the μ offspring. Then the best μ individuals according to $\prec_{Q^{(g)}}$ become the parents of generation $g + 1$.

2.2 Generational and Steady State MO-CMA-ES Algorithms

Algorithm 1 is a generic description of all MO-CMA-ES algorithms, where μ parents generate λ offspring. Borrowing Igel et al's notations [12], $a^{(g)}_i$ denotes some structure containing the i^{th} point x_i of the population at generation g together with its parameters related to the mutation (step-size, covariance matrix, average success rate, ...).

Lines 3-7 is the loop of offspring generation: at line 5 the parent is copied onto the offspring together with its parameters. It is mutated at line 6 (and evaluated). Then, depending on the success of the offspring (its performance compared to its parents'), the parameters of both individuals are updated between lines 8 and 16 (update of the covariance matrix in the case of successful offspring[1], and update of success rate and mutation step in any case).

The original MO-CMA-ES thus instantiates the generic Alg. 1 by taking $\lambda = \mu$ and having the parent selection on line 4 simply return its second argument ($i_k = k$, i.e., all parents generate exactly one offspring in turn).

In SS-MO-CMA-ES, $\lambda = 1$: only one offspring is generated at each generation. The parent selection (line 4 of Algorithm 1) proceeds by uniformly selecting either one from the μ parents (variant $(\mu+1)$-MO-CMA in the following); or one from the non-dominated parents (variant $(\mu_{\prec} + 1)$-MO-CMA in the following).

The survival selection (line 17) then amounts to replacing the worst parent x_w with the offspring if the latter improves on x_w according to $\prec_{Q^{(g)}}$, or discarding the offspring otherwise.

3 New Parent Selections for Steady-State MO-CMA-ES

After [12], the more greedy variant $(\mu_{\prec} + 1)$-MO-CMA outperforms all other variants on all unimodal problems. In contrast, on multi-modal problems such as ZDT4 and IHR4, $(\mu+1)$-MO-CMA performs better than $(\mu_{\prec}+1)$-MO-CMA [12], but it does not perform too well, and neither does the generational version of MO-CMA-ES, comparatively to other MOEAs.

These remarks naturally lead to propose more greedy parent selection operators within SS-MO-CMA-ES (line 4 of Alg. 1), in order to further improve its performances on unimodal problems, leaving aside at the moment the multimodality issue. A parent selection operator is based on i) a selection mechanism; and ii) a rewarding procedure (line 18). A family of such operators is presented in this section; the selection procedure either is based on a standard tournament selection (section 3.1), or inspired from the Multi-Armed Bandit paradigm (section 3.2). The rewarding procedures are described in section 3.3.

3.1 Tournament Selection

Standard tournament selection is parameterized from a tournament size $t \in \mathbb{N}$. Given a set X, t-tournament selection proceeds by uniformly selecting t individuals (with or without replacement) from X and returning the best one according to criterion at hand (here, the $\prec_{Q^{(g)}}$ criterion, see Eq. (1)). The parent selection procedure (line 4 of Alg. 1) thus becomes `TournamentSelection(`$Q^{(g)}$`)`.

The rewarding procedure (line 18 of Alg. 1) only computes for each parent its Pareto rank and Hypervolume contribution[2].

The Steady-State MO-CMA-ES using t-size Tournament Selection is denoted $(\mu +_t 1)$-MO-CMA in the following, or $(\mu +_t 1)$ for short. Parameter t thus

[1] The formulation of Algorithm 1 was chosen for its genericity. In practice however, only the surviving offspring will actually adapt their parameters; the update phase thus takes place after the survival selection (line 17).

[2] It is thus redundant with the Survival Selection (line 17), and can be omitted.

controls the selection greediness; the larger t, the more often points with high Hypervolume contribution will be selected on average.

3.2 Multi-Armed Bandit-Inspired Selection

Another parent selection procedure (line 4 of Alg. 1) inspired from the Multi-Armed Bandit (MAB) paradigm is described here. How to define the underlying rewards (line 18) will be detailed in next subsection.

The standard MAB setting considers several options, also called arms, each one with an (unknown but fixed) reward distribution [2]. The MAB problem is to find a selection strategy, selecting an arm $i(t)$ in each time step t and getting an instance of the corresponding reward distribution, such that this strategy optimizes the cumulative reward.

An algorithm yielding an optimal result has been proposed by Auer et al. [2]; this algorithm proceeds by selecting the arm which maximizes the sum of an exploitation term (the empirical quality, or average of rewards the arm has ever actually received) and an exploration term (enforcing that non-optimal arms be selected sufficiently often to enforce the identification of the truly optimal arm).

Considering that our setting is a dynamic one (as evolution proceeds toward the Pareto front), no algorithm with theoretical guarantees is available, and some heuristic adaptation of the above MAB algorithm is used:

1. The average reward of an arm (a parent) is replaced by its average reward along a time window of size w;
2. The exploration is enforced by selecting once every arm which i) occurs only once in the time window and ii) is about to disappear from the time window (it was selected w time steps ago);
3. In all other cases, the selection is on the exploitation side, and the arm with best average reward along the last w time steps is selected.

In summary, the MAB-like selection (line 4 of Alg. 1) always selects the parent with best average reward in the last w time steps, except for case 2 (a current parent is about to disappear from the time window). Parameter w thus controls the exploration strength of the selection. Experimentally however, the sensitivity of the algorithm w.r.t. w seems to be rather low, and w was set to 500 in all experiments (section 4).

3.3 Defining Rewards

This subsection describes the rewards underlying the MAB-like selection mechanism (line 18 of Alg. 1). A key related issue is how to share the reward between parents and offspring. On the one hand, if an offspring survives, it is better that some old parents and might thus be a good starting point for further advances toward the Pareto front. The offspring must thus inherit a sufficient fraction of its parent reward, to enable its exploitation. On the other hand, the reward of a parent should be high when it yields good-performing offspring, and in particular no reward should be awarded to the parent if the newborn offspring does not survive. Several reward indicators have been considered.

$(\boldsymbol{\mu + 1_{succ}})$. A first possibility is to consider boolean rewards. If an offspring makes it to the next generation, both the offspring and the parent receives reward 1. Formally:

$$r^{(g)} = 1 \text{ if } a_1'^{(g+1)} \in Q^{(g+1)}$$

Such boolean rewards entail a very greedy behavior. The newborn offspring, receiving 1 as instant reward, gets 1 as average reward over the time window; it will thus very rapidly (if not immediately) be selected at next parent. Likewise, its parent which already had a top average reward (it was selected), will improve its average reward and tend to be selected again.

$(\boldsymbol{\mu + 1_{rank}})$. A smoother reward is defined by taking into account the rank of the newly inserted offspring:

$$r^{(g)} = 1 - \frac{rank(a_1'^{(g+1)})}{\mu} \text{ if } a_1'^{(g+1)} \in Q^{(g+1)}$$

where $rank(a_1'^{(g+1)})$ is the rank of the newly inserted offspring in population $Q^{(g+1)}$ (using comparison operator $\prec_{Q^{(g)}}$ defined by Eq. (1); the top individual gets rank 0). Along this line, the reward ranges linearly from 1 (for a non-dominated individual with best hypervolume contribution) to 0. A newborn offspring will here be selected rapidly only if it makes it into the top-ranked individuals of the current population. The average reward of the parent can decrease if its offspring gets a poor rank, even if the offspring survives.

$(\boldsymbol{\mu + 1_{\Delta H_1}})$. Another way of getting smooth rewards is based on the hypervolume contribution of the offspring. Let us set the reward to 0 for dominated offspring (noting that most individuals are non-dominated in the end of evolution); for non-dominated offspring, one sets the reward to the increase of the total Hypervolume contribution from generation g to $g+1$:

$$r^{(g)} = \sum_{a \in Q^{(g+1)}} \Delta H(a, Q^{(g+1)}) - \sum_{a \in Q^{(g)}} \Delta H(a, Q^{(g)})$$

$(\boldsymbol{\mu + 1_{\Delta H_i}})$. In the early stages of evolution, many offspring are dominated and the above Hypervolume-based reward thus gives little information. A relaxation of the above reward, involving a rank-based penalization is thus defined. Formally, if k denote the Pareto rank of the current offspring, the reward is:

$$r^{(g)} = \frac{1}{2^{k-1}} \left(\sum_{ndom_k(Q^{(g+1)})} \Delta H(a, ndom_k(Q^{(g+1)})) - \sum_{ndom_k(Q^{(g)})} \Delta H(a, ndom_k(Q^{(g)})) \right)$$

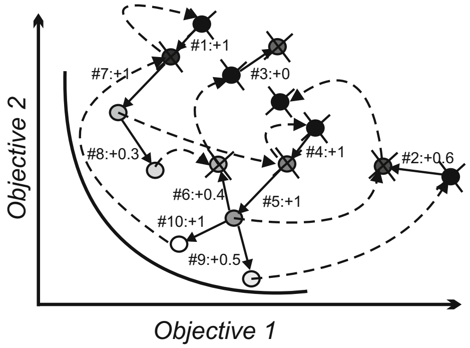


Fig. 1. Reward-based multi-objective optimization with bounded population. "#6:+0.4" means reward 0.4 on 6th iteration.

3.4 Discussion

The difficulty of associating a reward to a pair (parent, offspring) in Multi-Objective optimization is twofold. On the one hand, defining absolute indicators (e.g. reflecting some aggregation of the objective values) goes against the very spirit of MO. On the other hand, relative indicators such as above-defined must be taken with care: they give a snapshot of the current situation, which evolves along the population progress toward the Pareto front. The well-founded Multi-Armed Bandit setting, and its trade-off between Exploration and Exploitation, must thus be modified to account for non-stationarity.

Another difficulty is related to the finiteness of the population: while new arms appear, some old arms must disappear. The parent selection, e.g. based on the standard deterministic selection (Eq. (1)) is biased toward exploitation as it does not offer any way of "cooling down" the process. Such a bias is illustrated in Fig. 1. Let the population size of steady-state EMOA be 5, and consider a sequence of 10 evolution steps, generating 10 new points (oldest, resp. newest points are black resp. white). At each iteration the parent with best reward generates an offspring, then 6 points are compared using Eq. (1), and the worst point (crossed out) is eliminated. The instant parent reward reflects the quality of the offspring. Along evolution, some prospective points/arms are thus eliminated because they progress more slowly than others, although they do progress, due to the fixed population size. Expectedly, this bias toward exploitation adversely affects the discovery of multi-modal and/or disconnected Pareto front. We shall return to this issue in section 5.

4 Experimental Validation

This section reports on the validation of the proposed schemes, comparatively to the baseline MO-CMA-ES algorithms, detailing the experimental setting in section 4.1 before discussing the experimental evidence in section 4.2.

4.1 Experimental Setting

Algorithms. The experimentation involves:

- The steady-state MO-CMA-ES with tournament-based parent selection, where the tournament size t is set to 2 and 10 (respectively noted $(\mu +_2 1)$ and $(\mu +_{10} 1)$);
- The steady-state MO-CMA-ES with MAB-based parent selection, considering the four rewards described in section 3.2 (respectively noted $(\mu + 1_{succ})$, $(\mu + 1_{rank})$, $(\mu + 1_{\Delta H_1})$ and $(\mu + 1_{\Delta H_i})$);
- The baseline algorithms include the generational $(\mu + \mu)$-MO-CMA [10], and its steady-state variants $(\mu + 1)$-MO-CMA and $(\mu_{\prec} + 1)$-MO-CMA [12] (section 2.2).

All parameters of MO-CMA-ES are set to their default values [10] (in particular, $\mu = 100$); all algorithms only differ by their parent selection procedure. All reported results are based on 31 independent runs with at most 200,000 fitness evaluations, and median results are reported when the target precision was reached.

Problems. The well-known bi-criteria ZDT1:3-6 problems [17] and their rotated variants IHR1:3-6 [10] have been considered. Note however that the true Pareto front of all ZDT problems lies on the boundary of the decision space, which might make it easier to discover it. For the sake of an unbiased assessment, the true Pareto front is thus shifted in decision space: $x'_i \leftarrow |x_i - 0.5|$ for $2 \leq i \leq n$, where n is the problem dimension. The shifted ZDT problems are denoted sZDT. The set of recently proposed benchmark problems LZ09-1:5 [14] have also been used for their complicated Pareto front in decision space (Fig. 4).

Performance Measures. Following [13], the algorithmic EMO performance is measured from the hypervolume indicator I_H. Let P be a μ-size approximation of Pareto front and let P^* be the approximate μ-optimal distribution of optimal Pareto points [3]. The approximation error of the Pareto front is defined by $\Delta H(P^*, P) = I_H(P^*) - I_H(P)$.

Furthermore, to support an easy comparison of different algorithms across different problems, all results will be presented 'the horizontal way', i.e., reporting the number of function evaluations needed to reach a given precision. This procedure rigorously supports claims[3] such as *algorithm A is 2 times faster than algorithm B.*

4.2 Result Analysis

All empirical results are displayed in Table 1. These results show that the proposed algorithms generally outperform the baseline MO-CMA-ES approaches, with the exception of problems sZDT3, IHR6 and LZ09-5. A first general remark

[3] In opposition, a claim such as *Algorithm A can reach a precision 10 times smaller than algorithm B* is hard to assess when considering different problems.

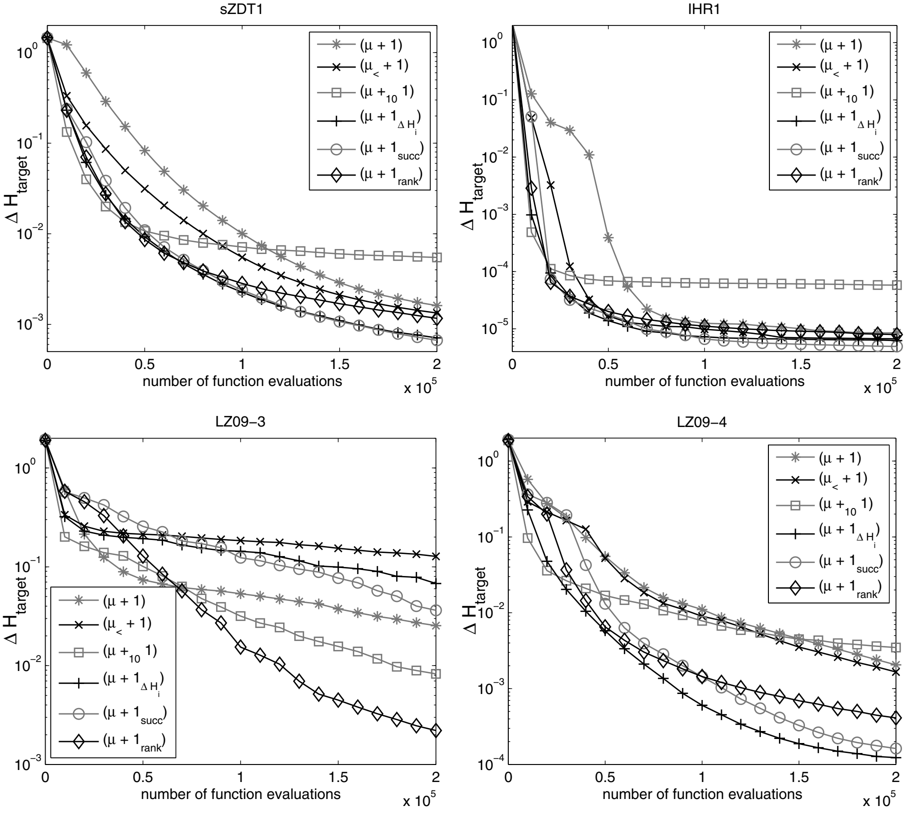


Fig. 2. On-line performances of baseline and proposed variants of steady-state MO-CMA-ES on sZDT1, IHR1, LZ09-3 and LZ09-4 problems (median out of 31 runs)

is that the steady-state variants of MO-CMA-ES outperform the generational one on unimodal benchmark problems; as already noted in [12], the greedier $(\mu_{\prec} + 1)$-MO-CMA is usually faster than the original steady-state on sZDT and IHR problems; in counterpart, it is too greedy on LZ09 problems (see below). Another general remark is that $(\mu + 1_{\Delta H_i})$-MO-CMA is usually more robust and faster than $(\mu + 1_{\Delta H_1})$-MO-CMA; this fact is explained as the former exploit a better informed hypervolume contribution based reward, considering also the contribution of dominated points.

The on-line performance of most considered algorithms on sZDT1, IHR1, LZ09-3 and LZ09-4 shows the general robustness of $(\mu + 1_{rank})$-MO-CMA (Fig.2, displaying $\Delta H(P^{*}, P)$ versus the number of function evaluations). The comparatively disappointing results of $(\mu + 1)$-MO-CMA on IHR1 are explained from the structure of the Pareto front, which includes an easy-to-find segment. This

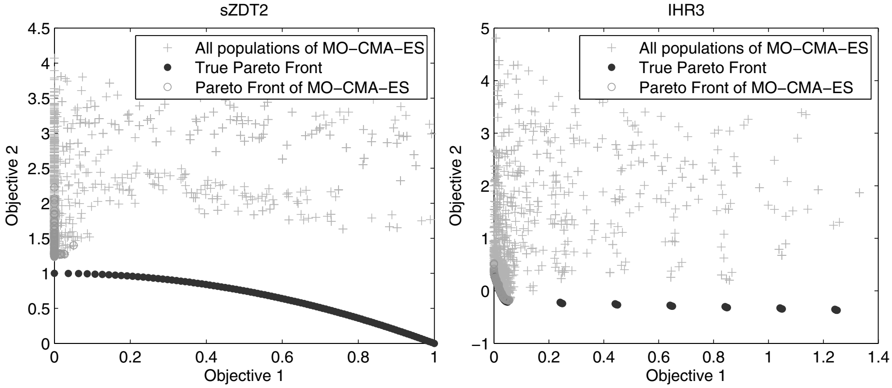


Fig. 3. Typical behavior of $(\mu + 1_{succ})$-MO-CMA on sZDT2 (left) and IHR3 (right) problems: premature convergence after 5,000 fitness function evaluations

segment can be discovered by selecting the extreme parent (in objective space), thus with probability $1/\mu$ within a uniform selection scheme. Quite the contrary, reward-based selection schemes quickly catch the fruitful directions of search.

The price to pay for this is depicted on Fig. 3, showing $(\mu + 1_{succ})$-MO-CMA on sZDT2 and IHR3 problems. On these problems, a premature convergence toward a small segment of the Pareto front is observed after circa 5,000 function evaluations. The interpretation provided for this premature convergence goes as follows. As one part of the Pareto front is easier to find than others, points aiming at this part quickly reach their goal; due to non-dominated sorting (and to the fixed population size), these eliminate other points, resulting in a very poor diversity (in objective space) of the population. This remark suggests that some additional diversity preserving technique should be used together with MO-CMA-ES; note that, even in the original MO-CMA-ES, a premature convergence is observed on IHR3.

LZ09 problems have also been considered because of their non-linear Pareto front in decision space (Fig. 4), contrasting with the linear Pareto front of all sZDT and IHR problems. The results depicted in Fig. 4 show that $(\mu + 1_{rank})$-MO-CMA better approximates the Pareto front than $(\mu + 1)$ and $(\mu +_{10} 1)$-MO-CMA, for all problems except LZ09-5. It is interesting to note that the results of $(\mu + 1_{rank})$-MO-CMA after 100,000 fitness evaluations match those of MOEA/D-DE after 150,000 fitness evaluations [14].

Overall (see also Table 1), $(\mu + 1_{rank})$-MO-CMA and $(\mu +_{10} 1)$-MO-CMA perform best on most problems, while $(\mu + 1_{\Delta H_i})$-MO-CMA is slightly more robust. Most generally, all greedy versions of MO-CMA-ES get better results on problems with a convex Pareto front; on problems with a concave or disconnected Pareto front, they suffer from premature convergence, entailed by a loss of diversity, due to non-dominated sorting and bounded population size.

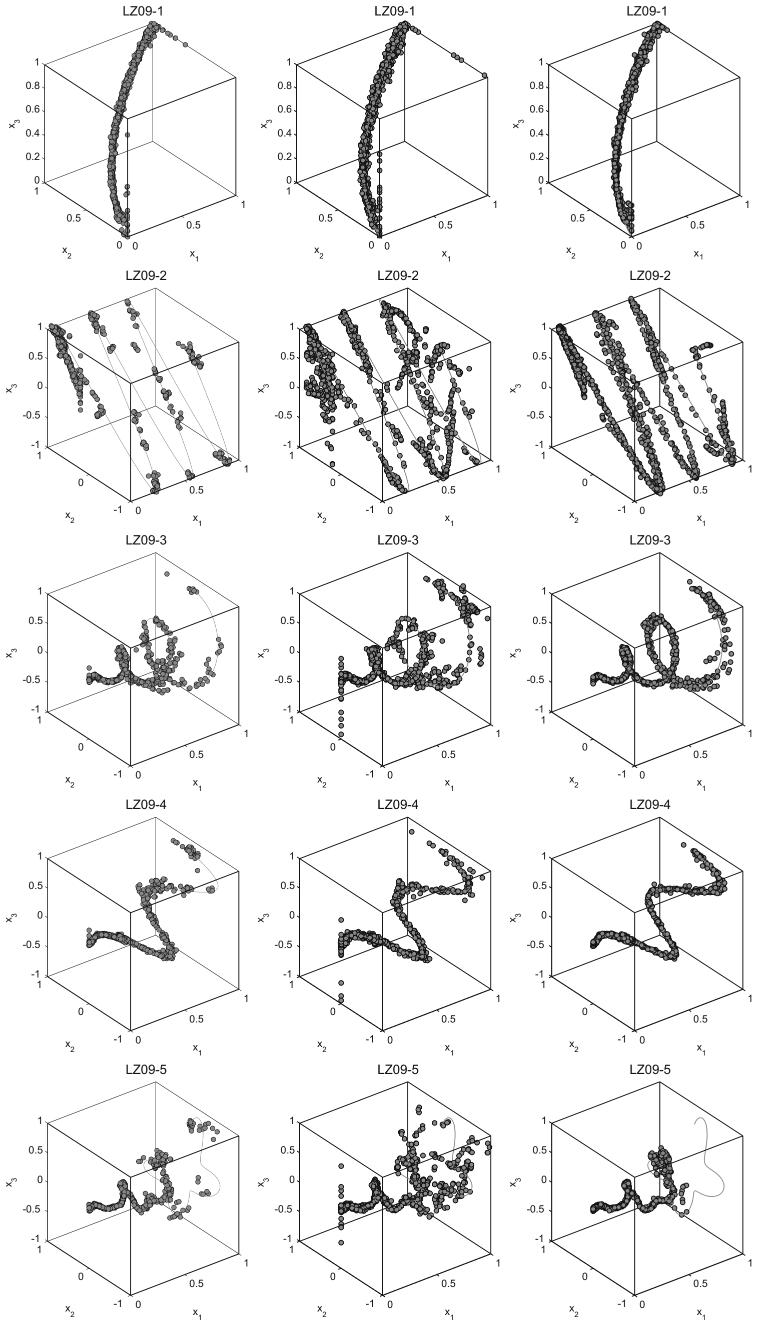


Fig. 4. Plots of all 10 populations found by $(\mu + 1)$-MO-CMA (left), $(\mu +_{10} 1)$-MO-CMA (center) and $(\mu + 1_{rank})$-MO-CMA (right) in the $x_1 - x_2 - x_3$ space on LZ09-1:5 after 100,000 function evaluations

Table 1. Comparative results of two baseline EMOAs, namely generational and steady-state MO-CMA-ES and several versions of steady-state MO-CMA-ES with different parent selection schemes. Median number of function evaluations (out of 31 independent runs) to reach ΔHtarget values, normalized by Best: a value of 1 indicates the best result, a value $X > 1$ indicates that the corresponding algorithm needed X times more evaluations than the best to reach the same precision.

ΔHtarget	1	0.1	0.01	1	0.1	0.01	1	0.1	0.01	1	0.1	0.01
		sZDT1			sZDT2			sZDT3			sZDT6	
Best	2500	12000	47000	2500	15000	59000	3000	18500	70000	4500	141200	.
$(\mu+\mu)$	7.3	4.3	2.2	8.6	4.3	2.1	5.5	3	1.5	8.2	**1**	.
$(\mu+1)$	6.5	3.9	2.1	7.6	3.9	2	5.1	2.8	1.4	7.2	1.1	.
$(\mu_{\prec}+1)$	1.5	2.2	1.7	**1**	2	1.5	1.3	1.8	1.2	**1**	.	.
$(\mu+_2 1)$	3.7	2.4	1.5	4.4	2.4	1.4	3.1	1.7	**1**	4.3	.	.
$(\mu+_{10} 1)$	1.2	**1**	1.1	1.4	**1**	1.3	**1**	**1**	.	1.3	.	.
$(\mu+1_{\Delta H_1})$	3.5	1.5	**1**	3.4	1.6	**1**	2.5	.	.	1.7	.	.
$(\mu+1_{\Delta H_i})$	1.7	1.3	**1**	1.8	1.3	**1**	1.1	.	.	1.5	.	.
$(\mu+1_{succ})$	1.2	1.7	1.1	1.6	.	.	**1**	.	.	2.1	.	.
$(\mu+1_{rank})$	**1**	1.4	**1**	1.4	.	.	**1**	.	.	1.7	.	.
		IHR1			IHR2			IHR3			IHR6	
Best	500	1500	6000	1500	4000	8500	1000	.	.	6000	.	.
$(\mu+\mu)$	8.4	8.8	6.9	6.4	4.8	3.3	8.2	.	.	5.6	.	.
$(\mu+1)$	7	7.3	6.7	5.6	4.1	2.9	7	.	.	5	.	.
$(\mu_{\prec}+1)$	**1**	**1**	3	**1**	1.6	1.7	**1**	.	.	**1**	.	.
$(\mu+_2 1)$	4	4.3	4	3.3	2.5	1.9	4	.	.	3	.	.
$(\mu+_{10} 1)$	2	1.6	1.1	**1**	**1**	**1**	**1**	.	.	**1**	.	.
$(\mu+1_{\Delta H_1})$	2	1.6	**1**	2	1.5	1.2	2.5	.	.	1.4	.	.
$(\mu+1_{\Delta H_i})$	2	2.3	**1**	1.3	1.3	1.1	1.5	.	.	1.2	.	.
$(\mu+1_{succ})$	2	2.3	2	5.3	2.7	1.7	1.5	.	.	1.9	.	.
$(\mu+1_{rank})$	2	2	1.5	1.6	1.7	1.3	1.5	.	.	1.6	.	.
		LZ09-1			LZ09-2			LZ09-3			LZ09-4	
Best	500	6000	17000	3500	144000	.	1500	35000	120500	1000	10000	40500
$(\mu+\mu)$	11.4	5.1	3.2	3.6	.	.	4.1	1.2	.	5.7	3.2	2.4
$(\mu+1)$	9	4.7	3	3.2	.	.	3.6	**1**	.	5	3.9	2.5
$(\mu_{\prec}+1)$	2	2.5	2.2	**1**	.	.	**1**	.	.	**1**	4.3	2.3
$(\mu+_2 1)$	6	2.8	1.9	2.2	.	.	2.3	1.7	.	3.5	2.7	1.9
$(\mu+_{10} 1)$	2	**1**	**1**	**1**	.	.	**1**	1.4	1.4	1.5	**1**	2
$(\mu+1_{\Delta H_1})$	9	2.1	1.5	2	.	.	1.6	5.6	.	2	1.8	**1**
$(\mu+1_{\Delta H_i})$	2	1.5	1.3	2.1	**1**	.	2	4.2	.	2.5	1.5	**1**
$(\mu+1_{succ})$	**1**	2.1	1.4	3.5	.	.	1.3	3.6	.	2	3.5	1.3
$(\mu+1_{rank})$	**1**	1.9	1.3	5.8	1.1	.	1.3	1.6	**1**	1.5	2.4	**1**
		LZ09-5										
Best	1500	19000	.									
$(\mu+\mu)$	3.4	1.6	.									
$(\mu+1)$	3.3	1.4	.									
$(\mu_{\prec}+1)$	**1**	.	.									
$(\mu+_2 1)$	2	**1**	.									
$(\mu+_{10} 1)$	**1**	1.7	.									
$(\mu+1_{\Delta H_1})$	1.3	.	.									
$(\mu+1_{\Delta H_i})$	1.6	1.9	.									
$(\mu+1_{succ})$	1.3	.	.									
$(\mu+1_{rank})$	**1**	.	.									

5 Conclusion and Perspectives

The goal and main contribution of the paper is to speed up MO-CMA-ES using new parent selection schemes, based on tournament and reward-based approaches inspired from the Multi-Armed Bandit framework, in order to quickly identify the most fruitful directions of search. Experiments on several bi-objective problems have shown a significantly speed-up of MO-CMA-ES on unimodal problems (for both generational and previous steady-state variants). However, the proposed approach results in a poor convergence on multi-modal multi-objective problems, or problems where some parts of the Pareto front are much easier to reach than others, such as IHR3 (Fig. 3, and discussion in sections 3.4 and 4.2).

These remarks open some perspectives for further research, aimed at preserving the benefits of parent selection schemes while addressing the premature convergence on multi-modal landscapes. A first perspective is to maintain the points located at the border of the already visited region, and to give them some chance to produce offspring as well although they are dominated. The question thus becomes to handle yet another exploitation vs exploration dilemma, and distribute the fitness evaluations between the current population and the borderline points; it also remains to extend the reward definition for the borderline points. Such an approach is similar in spirit to the so-called BIPOP-CMA-ES designed to overcome premature convergence within single-objective evolutionary optimization [7]; BIPOP-CMA-ES proceeds by maintaining one large population for exploration purposes, and a small one for fast and accurate convergence.

A second perspective is to design a more integrated multi-objective CMA-ES based algorithm, by linking the reward mechanism used in the parent selection and the internal update rules of CMA-ES. Indeed, the success rate used to control the $(1 + 1)$-ES evolution and the empirical success expectation used in $(\mu + 1_{succ})$-MO-CMA are strongly related. Further work will consider how to use the success rate *in lieu* of reward for parental selection, expectedly resulting in a more consistent evolution progress. Meanwhile, the CMA update rules might want to consider the discarded offspring (possibly weighting their contribution depending on their hypervolume contribution), since they might contain useful information even though they are discarded. Again, similar ideas have been investigated in the single objective case: the Active Covariance Matrix Adaptation [1] does use unsuccessful trials to update the distribution of mutation parameters. Some other recent proposals [5] might also help accelerating even further the MO-CMA-ES on separable functions: mirrored sampling systematically evaluates two symmetric points w.r.t. the mean of the Gaussian distribution, and sequential selection stops generating offspring after the first improvement over the parent.

Last but not least, the MO-CMA-ES and the proposed parent selection schemes must be analysed and compared with other state-of-the art MOEAs, specifically SMS-EMOA [4], the first algorithm to advocate the use of steady state within EMO to our best knowledge; it also proposed separable variation operators, resulting in excellent results comparatively to MO-CMA-ES on separable problems. How to extend these variation operators in the non-separable case, borrowing approximation ideas from [15] will be investigated.

References

1. Arnold, D.V., Hansen, N.: Active covariance matrix adaptation for the (1+1)-CMA-ES. In: Branke, J., et al. (eds.) ACM-GECCO, pp. 385–392 (2010)
2. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multi-armed bandit problem. Machine Learning 47(2-3), 235–256 (2002)
3. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In: FOGA, pp. 87–102. ACM, New York (2009)
4. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective Selection based on Dominated Hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)
5. Brockhoff, D., Auger, A., Hansen, N., Arnold, D.V., Hohm, T.: Mirrored sampling and sequential selection for evolution strategies. In: Schaefer, R., et al. (eds.) PPSN XI. LNCS, vol. 6238, pp. 11–20. Springer, Heidelberg (2010)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. IEEE TEC 6, 182–197 (2000)
7. Hansen, N., Auger, A., Ros, R., Finck, S., Posík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Branke, J., et al. (eds.) GECCO (Companion), pp. 1689–1696. ACM, New York (2010)
8. Hansen, N., Müller, S., Koumoutsakos, P.: Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evolution Computation 11(1) (2003)
9. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9(2), 159–195 (2001)
10. Igel, C., Hansen, N., Roth, S.: Covariance Matrix Adaptation for Multi-objective Optimization. Evolutionary Computation 15(1), 1–28 (2007)
11. Igel, C., Suttorp, T., Hansen, N.: A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In: Keijzer, M., et al. (eds.) GECCO 2006, pp. 453–460. ACM Press, New York (2006)
12. Igel, C., Suttorp, T., Hansen, N.: Steady-state selection and efficient covariance matrix update in the multi-objective CMA-ES. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 171–185. Springer, Heidelberg (2007)
13. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report TIK 214, ETH Zürich (2006)
14. Li, H., Zhang, Q.: Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Trans. Evolutionary Computation 13(2), 284–302 (2009)
15. Ros, R., Hansen, N.: A simple modification in CMA-ES achieving linear time and space complexity. In: Rudolph, G., et al. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 296–305. Springer, Heidelberg (2008)
16. Schumer, M., Steiglitz, K.: Adaptive step size random search. IEEE Transactions on Automatic Control 13, 270–276 (1968)
17. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)
18. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., et al. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)

On Sequential Online Archiving of Objective Vectors

Manuel López-Ibáñez[1], Joshua Knowles[2], and Marco Laumanns[3]

[1] IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
manuel.lopez-ibanez@ulb.ac.be
[2] School of Computer Science, University of Manchester, UK
j.knowles@manchester.ac.uk
[3] IBM Research, Zurich, Switzerland
mlm@zurich.ibm.com

Abstract. In this paper, we examine the problem of maintaining an approximation of the set of nondominated points visited during a multiobjective optimization, a problem commonly known as archiving. Most of the currently available archiving algorithms are reviewed, and what is known about their convergence and approximation properties is summarized. The main scenario considered is the restricted case where the archive must be updated online as points are generated one by one, and at most a fixed number of points are to be stored in the archive at any one time. In this scenario, the $\lhd$-monotonicity of an archiving algorithm is proposed as a weaker, but more practical, property than negative efficiency preservation. This paper shows that hypervolume-based archivers and a recently proposed multi-level grid archiver have this property. On the other hand, the archiving methods used by SPEA2 and NSGA-II do not, and they may $\lhd$-deteriorate with time. The $\lhd$-monotonicity property has meaning on any input sequence of points. We also classify archivers according to limit properties, i.e. convergence and approximation properties of the archiver in the limit of infinite (input) samples from a finite space with strictly positive generation probabilities for all points. This paper establishes a number of research questions, and provides the initial framework and analysis for answering them.

Keywords: approximation set, archive, convergence, efficiency preserving, epsilon-dominance, hypervolume, online algorithms.

1 Introduction

The convergence properties of large classes of multiobjective evolutionary algorithms were seriously considered for the first time in the late 1990s [18,9,17]. These papers laid the foundations for much of the analysis that has gone on to date, and showed that certain types of elitism combined with a certain type of generation process lead to convergence (in the limit) to a subset of the Pareto front (PF). Moreover, they indicated that, to a large extent, properties of a

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 46–60, 2011.

multiobjective stochastic search algorithm as a whole can be derived from separately considering properties of the generation process and properties of the elite-preserving mechanism.

Today, most multiobjective stochastic search algorithms are elitist in the sense of keeping an external *archive* (or memory) in order to capture the output of the search process. Because the set of minima visited may be very large in a multiobjective optimization process, it is common to bound the size of the archive. Thus, properties of the elite-preservation, or archiving, rules used to maintain bounded archives are of high interest to the community. Our aim in this paper is to elucidate, in one place, some of the properties of existing archiving algorithms that keep at most a fixed maximum number of points to approximate the PF. We restrict our attention to sequential archiving of points that arrive one-by-one, but consider a number of differently motivated algorithms for this setting. We consider archiving algorithms aimed only at convergence (similar to AR1 [17]), algorithms aimed mostly at 'diversity'[1] (derived from the elite population update rules of SPEA2 [19] and NSGA-II [7]), algorithms that consider overall approximation quality (epsilon dominance-based [16], grid-based [13], and based on maximizing hypervolume [11,1]), including a relatively new proposal called multi-level grid archiving [15]. We review the properties of these archiving algorithms and illustrate them empirically.

2 Preliminaries

We are concerned with vectors (points) in finite, multidimensional objective spaces. Let $Y \subset \mathbb{R}^d$ be a finite objective space of dimension $d > 1$. An order relation on Y may be defined as follows: $y \prec y'$ iff $\forall i \in 1, \ldots, d$, $y_i \leq y'_i$ and $y \neq y'$. Thus $\prec$ is a strict partial order on Y. Instead of $y \prec y'$ we may also write y *dominates* y'. The set of minimal elements of Y may be defined as

$$Y^* := \min(Y, \prec) = \{y \in Y, \nexists y' \in Y, y' \prec y\}.$$

The set Y^* is called the *Pareto front* (PF). Any other set $P \subseteq Y$ with the property $P = \min(P, \prec)$ will be called a *nondominated set.*

We are interested in finding approximations of the set Y^* of cardinality at most N. Such approximation sets are also partially ordered when we extend the definitions of dominance to pairs of sets as follows. Let P be a nondominated set. A point $y \notin P$ is *nondominated* w.r.t. P iff $\nexists y' \in P, y' \prec y$. Let P and Q be two nondominated sets. Then $P \lhd Q$ iff $\min(P \cup Q, \prec) = P \neq Q$.

2.1 Optimal Approximation Sets of Bounded Size

The partial order on sets defined by $\lhd$ gives the primary *solution concept* for determining an optimal approximation set of size at most N, as follows:

[1] The term "diversity" has no fixed definition in the literature, but it can refer to the evenness of the spacing between points and/or the extent of the nondominated set.

Definition 1 (Optimal Approximation Set of Bounded Size). *If $A \subseteq Y$ is a nondominated set, $|A| \leq N$, and $\nexists B \subseteq Y, |B| \leq N, B \lhd A$, then A is an* optimal approximation set of bounded size N *of Y^*.*

This solution concept derives from the dominance partial order only, but is in general not sufficient to guide a search or archiving process alone. We are now used to the notion of evaluating approximation sets with performance indicators, and using performance indicators to define other solution concepts that are *compatible* with dominance (see below), i.e. they are *refinements* [21] of it, that may be more suitable for guiding an archiving algorithm.

2.2 Compatibility of Performance Indicators

Let J be the set of all nondominated subsets of Y. A *unary performance indicator* $I\colon J \rightarrow \mathbb{R}$ is a mapping from the set J to the real numbers. Assuming that the indicator's value is to be minimised, we can define compatibility of I with respect to $(J, \lhd)$. If $\nexists A, B \in J$, such that $A \lhd B$ and $I(A) \geq I(B)$ then I is a *compatible* indicator [10,22]. Analogously, a *weakly compatible indicator* can be defined by replacing $I(A) \geq I(B)$ with $I(A) > I(B)$ in the statement above.

Hypervolume indicator. The hypervolume indicator HYP(A) [20] of an approximation set A (originally called the $\mathcal{S}$ metric in the literature) is the Lebesgue integral of the union of (hyperectangular, axis-parallel) regions dominated by the set A and bounded by a single d dimensional reference point that must be dominated by all members of the true PF. The indicator's value should be maximized. The compatibility of the indicator [12,22] is behind its importance as a performance assessment method and as a means of guiding search and archiving algorithms.

Additive ϵ indicator. A point y is said to be weakly ϵ-dominated by a point y' iff $\forall i \in 1, \ldots, d,\ y'_i \leq y_i + \epsilon_i$. The unary epsilon indicator $\epsilon_{add}(A)$ of an approximation set A is defined as the minimum value of ϵ such that every point in Y^* is weakly ϵ-dominated by an element of A. This indicator has been shown to be weakly compatible with the $\lhd$-relation on sets [22] following the proposal of ϵ-dominance as a means of evaluating and obtaining approximation sets [16].

3 Archivers, Convergence and Approximation

Similarly to earlier papers [16,13,6], the setting we consider is that some generation process is producing a *sequence* of points (objective vectors) $\langle y^{(1)}, y^{(2)}, \ldots \rangle$, and we wish to maintain a subset of these minima in an archive A of fixed maximum size, $|A| \leq N$. We denote by A_t the contents of the archive after the presentation of the t-th objective vector. An *archiver*, i.e., an archiving algorithm for updating A with y, is an online algorithm [2] as it has to deal with a stream of data with no knowledge of future inputs. Knowles and Corne [13] previously showed that this online nature of the task means that no archiver can guarantee to have in its archive $\min(N, |Y_t^*|)$ where Y_t^* is the set of minimal elements of the input sequence up to a time t. A corollary of this, not previously stated explicitly, is that no online archiver of bounded size can deliver an 'optimal approximation set of bounded size' even in the weak sense of Definition 1.

3.1 Convergence and Approximation Definitions

When analysing an archiver's behaviour, we may be interested in how it performs in general input sequences of finite length, where points do not necessarily appear more than once in the sequence. This scenario models a one-pass finite sample of the search space. Or we may be interested in sequences where every point is seen an infinite number of times [17]. When considering the one-pass setting, we wish to know whether the archive is always a good approximation of the input sequence (at every time step). When considering the behaviour on points drawn indefinitely from a finite space, we wish to know whether convergence ever occurs (does the archive stop changing eventually?), and if so, what kind of approximation set is obtained, i.e. what is the archiver's limit behaviour. The following definitions expand on these ideas. The first four are properties that apply to one-pass settings (which also imply they are limit properties, too). Two limit-behaviour definitions follow.

Definition 2 ($\subseteq Y^*$). *No point in the archive is dominated by a point in the input sequence:* $\forall t,\ \forall y \in A_t, y \in Y_t^*$.

Definition 3 (diversifies). *An archiver is efficiency preserving [9] when full, if* $\forall t,\ |A_t| = N,\ y \in A_{t+1}$ *iff* $\exists y' \in A_t, y \prec y'$*. That is, it cannot accept points outside of the region dominating the current archive, thus limiting the diversity of points in the archive. We say that an archiver* without *this property* diversifies *by discarding a nondominated point from the archive to accept the new one.*

Definition 4 (negative efficiency-preserving [9]). *There does not exist a pair of points* $y \in A_t$ *and* $y' \in A_v,\ t < v$ *such that* y *dominates* y'*. Let an archiver that does not have this property be said to* deteriorate.

Definition 5 ($\lhd$-monotone). *There does not exist a pair of sets* A_t *and* A_v*,* $t < v$ *such that* $A_t \lhd A_v$*. Let an archiver that does not have this property be said to* $\lhd$-deteriorate.

Definition 6 (limit-stable). *For any sequence consisting of points drawn indefinitely with a strictly positive probability from a finite set, there exists a* t *such that* $\forall v > t,\ A_t = A_v$*. That is, the archive set converges to a stable set in finite time.*

Definition 7 (limit-optimal). *For any sequence consisting of points drawn indefinitely with a strictly positive probability from a finite set, the archive will converge to an optimal bounded archive (see Definition 1).*

Table 1 summarises the properties of the eight archivers in terms of Definitions 2–7. An illustration of some of these concepts is given in Fig. 1.

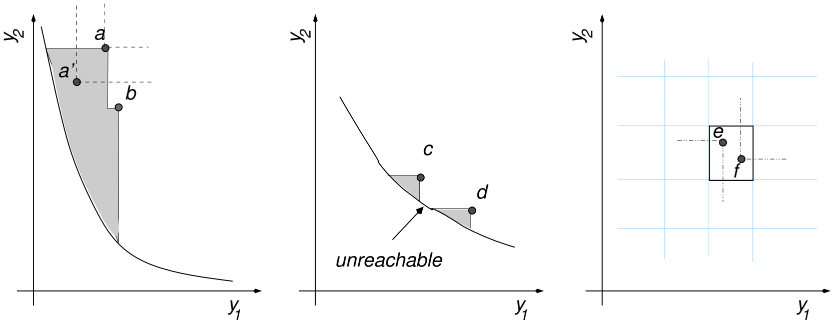


Fig. 1. Illustrations of some convergence concepts. (Left) Consider that $\{a, b\}$ is an archive; then an *efficiency-preserving* archiver may only accept a point in the (dominating) shaded region. If it accepts a' (removing a) this is also *negatively efficiency preserving* because the total region dominated is a superset of the region dominated by a, as indicated by the dashed lines. (Middle) Consider a different archive represented by c and d. In this case, a negatively efficiency preserving archiver can cause points to be unreachable, since only points within either of the shaded regions can now be accepted (adapted from Hanne [9]). (Right) Points e and f illustrate how the ϵ-Pareto archiver manages to guarantee that only Pareto optimal points are in its final archive. Two points in the same box cannot co-exist so one will be rejected from the archive. Let us say it is f. Only points which dominate e are allowed to populate the box in the future. Since the intersection of the region dominating e and the region dominated by f is empty, this ensures that, although f is no longer in the archive, no point dominated by f ever enters the archive.

3.2 Basic Archiver Pattern

Six of the eight archivers we study (all except for the two ϵ-based ones [16]) follow the scheme of Algorithm 1. These archivers describe a class called "precise" [6]. It is helpful for the later analysis of each individual archiver to observe the properties of Rules 1 and 2 (see Algorithm 1). Rule 1 is *efficiency-preserving* [9], which means that the region that contains points that dominate the archive after application of the rule is a subset of this region before the rule was applied. The rule is also *negative efficiency preserving* (Ibid.), which means that the region dominated by the archive after application of the rule is a superset of this region before. Rule 2 on the other hand is just *negative efficiency preserving.* For other properties of the algorithms described below, see Table 1.

3.3 Unbounded Archive

Trivially, the archiver yields the Pareto front of the input sequence. Although it is negative efficiency preserving [9] (Def. 4), it does not suffer from the curse of *unreachable points* (Ibid.) because these only occur when the set of points is also size limited.

Table 1. Types of convergence behaviour displayed by the archivers, and broad indication of time complexity for archive update. P denotes polynomial in N and d, and $E(d)$ expresses exponential in d.

Archiver	$\subseteq Y^*$	diversifies	negative efficiency-preserving	$\lhd$-monotone	limit-stable	limit-optimal	Complexity
Unbounded	+	+	+	+	+	+	P
Dominating	-	-	+	+	+	+	P
ϵ-approx	-	+	-	+	+	-	P
ϵ-Pareto	+	+	+	+	+	-	P
NSGA-II	-	+	-	-	-	-	P
SPEA2	-	+	-	-	-	-	P
AGA	-	+	-	-	-	-	P
AA$_S$	-	+	-	+	+	+	$E(d)$
MGA	-	+	-	+	+	+	P

Algorithm 1. Basic Archiver Pattern

```
Input: $A_{t-1}, y$
if $\exists y' \in A_{t-1}, y' \prec y$ then
    $A_t \leftarrow \min(A_{t-1} \cup \{y\})$    // Rule 1
else if $|\min(A_{t-1} \cup \{y\})| \leq N$ then
    $A_t \leftarrow \min(A_{t-1} \cup \{y\})$    // Rule 2
else
    $A_t \leftarrow \mathsf{filter}(A_{t-1} \cup \{y\})$    // $\mathsf{filter}(\cdot)$ returns a set of size $N$
end if
Output: $A_t$
```

3.4 Dominating Archive

The simplest way to achieve an archive of fixed maximum size is to implement the Basic Archiver with the $\mathsf{filter}(\cdot)$ function that just returns A_{t-1}. In other words, this archiver admits only dominating points whenever it is at full capacity. This archiver, when connected to a suitable sequence-generating process, is similar to the AR1 algorithm [17]. Due to the use of Rules 1 and 2 in combination (only), the archiver is negative efficiency preserving. Two corollaries of this are that the archive cannot *deteriorate* [9] (Def. 4), and it will always contain a subset of the Pareto front of the input sequence. However, the archiver gives no guarantee of approximation quality, and, in practice, especially for small N, it will tend to an almost *efficiency preserving* behaviour where it shrinks into a small region of the Pareto front. The archive may also contain points that are not Pareto optimal in the input sequence (even though deterioration does not occur), because $|A|$ may fall below N (due to Rule 1) and points dominated in Y may be accepted because the dominating point in Y^* was previously rejected entry into the archive due to rule $\mathsf{filter}(\cdot)$, at an earlier timestep when the archive was full.

3.5 Adaptive ϵ-Approx Archiving

The ϵ-approx archiver [16] does not follow our previous pattern. In this algorithm, a point is accepted only if it is not ϵ-dominated by an archived point. If it is accepted, then $A_t \leftarrow \min(A_{t-1} \cup \{y\})$, as usual. For fixed ϵ, it was shown that the archive is always an ϵ-approximate set of the input sequence of finite size (but not limited to any fixed value).

Laumanns et al. [16] also describe an adaptive scheme in order to allow a user to specify a maximum archive size N, rather than an ϵ value. However, this scheme often results in too large values of ϵ with the result that too few points are archived (e.g. compared to AGA) [14]. Hence, although the archiver is $\triangleleft$-monotone, it is not limit-optimal. Other properties are summarised in Table 1.

3.6 Adaptive ϵ-Pareto Archiving

The second archiver in [16] uses the idea that objective space can be discretized, via ϵ, into equivalence classes called 'boxes', so that every objective vector belongs to precisely one box. Within a box, only one point is allowed to exist in the archive, and the update rule within a box allows only a dominating point to replace the incumbent (see Fig. 1). This scheme guarantees that every point in the archive is Pareto optimal wrt the input sequence. This is the only archiver here that has this property and maintains a size-bounded archive.

Similarly to the ϵ-approximate archiver, a scheme to adapt ϵ on the fly was also proposed in [16] so that an archive limited to N points could be obtained. But this adaptation scheme does not facilitate reducing ϵ if it starts or becomes too large, with the result that the archiver keeps too few solutions, preventing it from being limit-optimal.

3.7 NSGA-II Archiver

The NSGA-II algorithm [7] assigns different selective fitness to nondominated points on the basis of their *crowding distance*, a coarse estimate of the empty space that surrounds a point. Our NSGA-II archiver follows the scheme of the Basic Archiver (Algorithm 1), and implements the $\mathsf{filter}(\cdot)$ function by removing the point with minimum crowding distance [7].

Since crowding distance is independent of dominance, no convergence guarantees can be made. It does not yield a subset of the nondominated points from the input sequence, in general. More importantly, the archive may $\triangleleft$-deteriorate (Definition 5), and we later show this empirically in Section 4.4. Moreover, even on a sequence constructed from an indefinite random sampling of a finite space, the archive may never settle to a stable set.

3.8 SPEA2 Archiver

The external population update of SPEA2 [19] was designed to prevent some of the regression and oscillation observed in the original SPEA. Our SPEA2 archiver follows the scheme of the Basic Archiver (Algorithm 1), but uses the distance to the k-nearest neighbour as the density measure in the $\mathsf{filter}(\cdot)$ function, as is used in SPEA2 for update of the external population.

The SPEA2 archiver has similar properties to NSGA-II archiver in terms of convergence and approximation: The archive can $\lhd$-deteriorate, and the archiver is not limit-stable. Moreover, we show in Section 4.3 that even for a sequence of all Pareto-optimal points, the diversity measure of SPEA2 may lead to very poor approximation quality.

3.9 Adaptive Grid Archiving (AGA)

Adaptive grid archiving uses a grid over the points in objective space in order to estimate local density. Its $\mathsf{filter}(\cdot)$ rule in the instantiation of Algorithm 1 is

$$A_t \leftarrow A_{t-1} \cup \{y\} \setminus \{y_c \in C\}$$

where y_c is a point drawn uniformly at random from C, the set of all the vectors in the "most crowded" grid cells, excluding any points that are a minimum or maximum on any objective within the current archive.

The archive rule is neither negatively efficiency preserving nor efficiency preserving, so AGA can deteriorate. Neither is it $\lhd$-monotone, a more serious problem. Only under special conditions (the grid cells are correctly sized and the grid stops moving) does a form of approximation guarantee become possible [11].

3.10 Hypervolume Archiver AA_S

This archiver was first proposed by Knowles [11] and follows the pattern of Algorithm 1, with the $\mathsf{filter}(\cdot)$ rule:

$$A_t \leftarrow \arg\max_{A \in \mathcal{A}_N} \{\mathrm{HYP}(A)\},$$

where $\mathcal{A}_N$ is the set of all subsets of $A_{t-1} \cup \{y\}$ of size N. In the one pass scenario, greedily removing the least-contributor does not ensure that the hypervolume is maximized over the whole sequence [4]. In Section 4.3, we provide an example where AA_S clearly does not maximize the hypervolume. This also means that it is neither negative efficiency preserving nor efficiency preserving: A point in the archive may be dominated by one that was previously in the archive, i.e., it may deteriorate. However, since the hypervolume never decreases, the archiver is $\lhd$-monotone (Definition 5).

The behaviour in the limit fulfills the solution concept (Definition 1), i.e. it is limit-optimal. The archive will be a set of $\min(N, |Y^*|)$ Pareto-optimal points after sufficiently long time, since if a set of size N has its maximum hypervolume value (out of all sets of such size) then all the points are Pareto optimal [8, Theorem 1].

Bringmann and Friedrich [5] have proved that hypervolume approximates the additive ϵ indicator, converging quickly as N increases. That is, sets that maximize hypervolume are near optimal on additive ϵ too, with the 'gap' diminishing as quickly as $O(1/N)$.

Updating the archive may be computationally expensive for large d and N. But despite the intractability of finding the point contributing least to the hypervolume in a set, approximation schemes may be good enough in practice [3].

3.11 Multi-level Grid Archiving (MGA)

The multi-level grid archiving (MGA) algorithm [15] can be thought of as combining principles from AGA and the ϵ-Pareto archiver. It was designed from the outset to maintain at most N points, achieving this by using a hierarchical family of boxes (equivalence classes) of different coarseness over the objective space. Specifically, when comparing solution at coarseness level $b \in \mathbb{Z}$, the components y_i of their objective vectors $y \in \mathbb{R}^d$ are mapped to (integral) values $\lfloor y_i \cdot 2^{-b} \rfloor$ to define its *box index vector* at level b.

The archiver follows the pattern of Algorithm 1. Its $\mathsf{filter}(\cdot)$ rule works by first determining the smallest level b where at least one of the $N+1$ candidates' box index vector is weakly dominated. The new candidate y is rejected if it belongs to the points that are weakly dominated at this level b; otherwise an arbitrary solution from this set is deleted. Through this adaptive determination of the right coarseness level for comparison, the behaviour observed in the ϵ-archivers of ending up with too large an ϵ value can be avoided, as we later show experimentally in Section 4.1.

The archiver is neither negatively efficiency preserving nor efficiency preserving, which means that it does *not* guarantee that its archive contains only Pareto points of the input sequence. We provide an example of this in Section 4.5. Nevertheless, it is shown in [15] that any archive update strictly increases a unary performance indicator compatible with dominance (i.e., it is $\lhd$-monotone, see Def. 5), like the hypervolume archiver AA_S. However, unlike the AA_S, MGA does not calculate this unary indicator explicitly, which makes it computationally more tractable than AA_S. In particular, its time complexity is $O(d \cdot N^2 \cdot L)$, where L is the length of the binary encoded input, therefore polynomial.

4 Empirical Study

Despite their crucial importance in the quality of MOEAs, there is surprisingly little experimental work on the behaviour of different archivers [16,13,6]. We provide in this section experiments that confirm the observations in the previous sections, and illustrate some properties of popular archivers that have not been described in the literature.

We have implemented the various archiving algorithms in C++ within a common framework. We make available the initial version of this framework at `http://iridia.ulb.ac.be/~manuel/archivers` in order to help future analysis. We plan to extend this framework in the future with other archivers found in the literature.

In this section, we empirically analyse the reviewed archiving algorithms. In order to focus on the properties of the algorithms, we study the performance

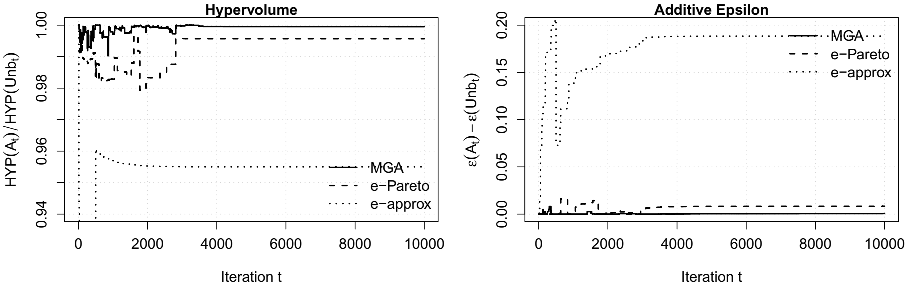


Fig. 2. Small PF (2D) $N = 10$

of the algorithms when presented with particular sequences of points. The sequences studied have been generated in order to highlight some characteristics of the algorithms.

We evaluate the quality of the algorithms with respect to the hypervolume and unary ϵ-indicator. In all sequences, we run the unbounded algorithm and keep the Pareto front at each iteration of the sequence. In the case of the additive ϵ measure (ϵ_{add}), the reference set is the optimal PF (which is the final unbounded archive). Then, for each archiver and at each iteration, we calculate $\epsilon_{add}(A_t) - \epsilon_{add}(\mathsf{Unbounded}_t)$. Similarly, for the hypervolume we calculate the reference point over the final unbounded Pareto front as

$$r_i = \max f_i + ((1 + (1/(N - 1))) \cdot (\max f_i - \min f_i)).$$

Then, we calculate the ratio $\mathrm{HYP}(A_t)/\mathrm{HYP}(\mathsf{Unbounded}_t)$, for each iteration t of the input sequence.

In all sequences, the objective functions are to be minimized, without loss of generality since the sequences are finite, we could always transform them into an all-positive maximization problem and the results will stand.

4.1 MGA Addresses Key Weakness of ϵ-Archivers

In both ϵ-approximate and ϵ-Pareto algorithms, the ϵ may become arbitrarily large with respect to the extent of the Pareto front. Knowles and Corne [14] showed that this occurs, for example, when the initial range of objective values is much larger than the actual range of the Pareto front. In that case, the initial estimate of ϵ is much larger than actually needed, but since ϵ cannot decrease, the algorithms end up accepting fewer points than N. This situation occurs even with a small initial estimate of $\epsilon = 0.0001$, as we use in the experiments here. We ran experiments on two sequences proposed by Knowles and Corne [14], of length 10 000 and dimensions 2 and 3, respectively. Fig. 2 and Fig. 3 show that these sequences are not a problem for MGA. Moreover, while MGA is able to maintain an archive size of $|A| = 20$, ϵ-approximate and ϵ-Pareto only keep 2 and 1 solutions respectively just after 4 000 iterations until the end.

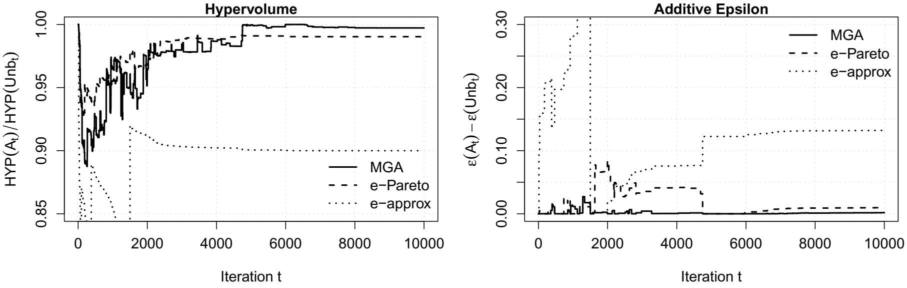
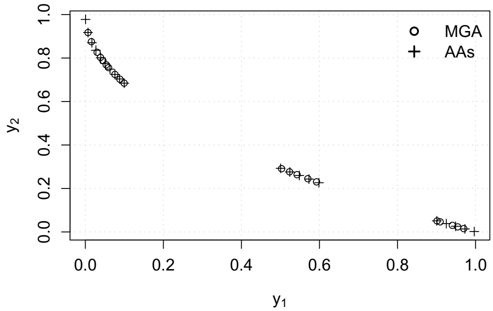


Fig. 3. Small PF (3D) $N = 10$



	HYP	ϵ_{add}
AA$_s$	0.9993076	0.009689
MGA	0.9929917	0.013613

Fig. 4. MGA vs. AA$_s$ for clustered points, $N = 20$

4.2 MGA vs. AA$_s$ for Clustered Points

We use a clustered sequence of 900 points in two dimensions to show the different final sets archived by AA$_s$ and MGA. Fig. 4 shows that AA$_s$ keeps the extremes of each cluster, whereas MGA points are not sparsely distributed within each cluster. The result is that AA$_s$ obtains better value in all performance indicators.

4.3 Fast Degradation of the SPEA2 Archiver

We illustrate how the quality of SPEA2 archiver can degrade very fast if points are added in the extreme of the Pareto front. We generate a sequence of 2 000 nondominated points in a straight line, sorted in increasing order of their first dimension. The top plots in Fig. 5 show that the quality of the archive stored by SPEA2 archiver degrades very rapidly as the sequence progresses. What is happening is that SPEA2 keeps the $N - 1$ initial solutions plus the new extreme, which replaces the old extreme. Therefore, at every step, the gap between the new extreme and the $N - 1$ initial solutions increases further. The final archives are shown in the bottom plot of Fig. 5. All but one solutions archived by SPEA2 are clustered in the left-most extreme of the PF.

The plot also shows that neither MGA nor AA$_s$ obtain a perfect approximation, which for this particular sequence would mean a uniformly distributed archive. Since they do not have knowledge about the real range of the PF, they

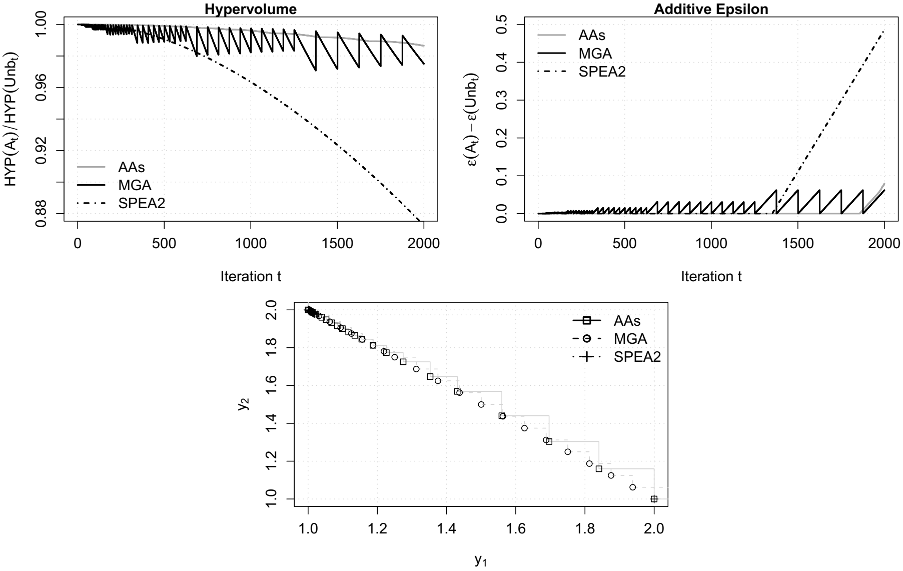


Fig. 5. Increasing extremes (2D) affects SPEA2 performance, $N = 20$

cannot accurately decide when to keep a solution close to the moving extreme. Nonetheless, MGA and AA$_s$ do not suffer the fast degradation in approximation quality shown by the SPEA2 archiver.

4.4 The NSGA-II Archiver ◁-Deteriorates

It is possible to construct a sequence of points such that the NSGA-II archiver removes points from its archive that are Pareto-optimal and includes points that are dominated in such a way that the archived set may be dominated by a previously archived set, and therefore, we say that the quality of the archive has ◁-deteriorated over time (Definition 5). Fig. 6 shows the final archive stored by Unbound, AA$_s$, MGA and NSGA-II. Except for the extremes, the rest of the final archive stored by NSGA-II archiver is dominated by solutions stored in previous archives. In fact, for this sequence, the archive at step $t = 58$ is dominated by the archive at step $t = 56$. It is possible to construct different sequences that show the same behaviour for the SPEA2 archiver.

4.5 MGA Is Not Negatively Efficiency Preserving

In general, MGA is not negatively efficiency preserving, since the final archive may contain points that are dominated by points that were previously in the archive and deleted. This is exemplified in the sequence shown in Fig. 7 for $N = 4$. In this sequence, MGA deletes point d after archiving point e. Then, a and b become dominated by f, and g is accepted into the archive, despite it is dominated by d. A sequence showing that AA$_s$ is not negatively efficiency preserving can be constructed by placing g such that it is dominated only by c.

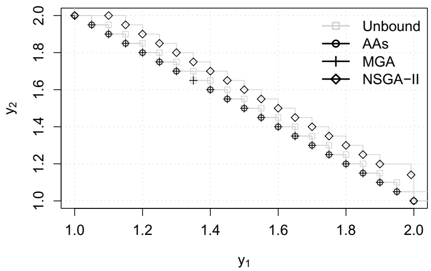


Fig. 6. NSGA-II may $\lhd$-deteriorate over time, $N = 20$

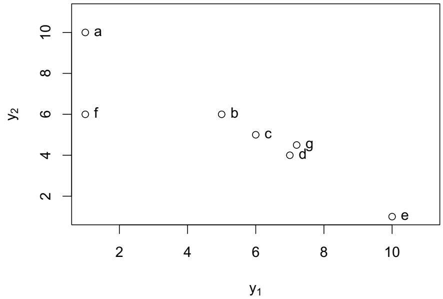


Fig. 7. The final archive of MGA with $N = 4$ is $\{c, e, f, g\}$, which is not negatively efficient preserving

5 Conclusions

In this paper we have examined the problem of keeping a bounded size approximation of the Pareto front of a set of points in a d-dimensional (objective) space, when the elements of the set are only accessed one-by-one. This models the *archiving* process of keeping an elite population or bounded size "best-so-far" outcome in many multi-objective optimizers.

Earlier works on this problem have dealt with algorithms designed to be stand-alone archivers, such as AGA and ϵ-based archivers. However, the diversity mechanisms employed by popular MOEAs are also archiving algorithms. In this paper, we have proposed a classification of both kinds of archivers, and the recently proposed MGA, according to a number of properties not considered before for this problem (summarised in Table 1). In particular, we differentiate between negative efficiency preservation and $\lhd$-monotonicity, and identify two classes of archivers, one based on compatible indicators (hypervolume-based AA_s and the new MGA), and another based on diversity mechanisms (SPEA2, NSGA-II and AGA). This allows us to understand more precisely why the former class of archivers have better convergence properties than the latter class, even when points are seen just once. The former cannot $\lhd$-deteriorate, even if a single point in the archive can be dominated by one that was previously in the archived set. This classification raises the question as to whether there may exist an archiver of the same class as MGA and AA_s that is also negative efficiency preserving.

In addition, our experiments have shown that the recently proposed MGA addresses the key weakness of the earlier ϵ-based archivers, however, at the cost of losing the guarantee of only archiving Pareto-optimal solutions. As a final observation, we did not find an absolute winner, but a tentative assessment is that AA_s often produces better results with respect to hypervolume, whereas MGA often obtains the best ϵ-measure values.

This paper has shown that the archiving problem is far from being well-understood, and we have left open a number of questions. First, we have only examined artificial sequences, designed to show the properties defined here. An interesting extension is to assess the typical performance of the archivers on multiple runs for various simple geometric sequences in varying dimensions, and also from points coming from stochastic search on standard benchmark problems. Second, we have limited ourselves to one-by-one archiving of points and (mostly) a one-pass setting. We know that updating the archive with more than one point simultaneously cannot be a worse approach, and for hypervolume it has already been shown to be superior. Therefore, understanding how the properties defined here extend to other update scenarios is an open research question. Third, we plan to extend this work to other archivers found in the literature, and to foster that project we also provide the archivers and artificial sequences used here to the community.[2] Fourth, we plan to use competitive analysis techniques from the field of online algorithms to obtain worst-case bounds, in terms of a measure of "regret" for archivers. Finally, after highlighting some weaknesses of existing archivers, we ask whether designing a better archiver is possible, and what trade-offs exist in its design.

Acknowledgments. Manuel López-Ibáñez acknowledges support from the FRFC project "*Méthodes de recherche hybrides pour la résolution de problèmes complexes*".

References

1. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)
2. Borodin, A., El-Yaniv, R.: Online computation and competitive analysis. Cambridge University Press, New York (1998)
3. Bringmann, K., Friedrich, T.: Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 6–20. Springer, Heidelberg (2009)
4. Bringmann, K., Friedrich, T.: Don't be greedy when calculating hypervolume contributions. In: Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA), pp. 103–112 (2009)
5. Bringmann, K., Friedrich, T.: The maximum hypervolume set yields near-optimal approximation. In: Pelikan, M., Branke, J. (eds.) GECCO 2010, pp. 511–518. ACM Press, New York (2010)
6. Corne, D., Knowles, J.D.: Some multiobjective optimizers are better than others. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), vol. 4, pp. 2506–2512. IEEE Press, Piscataway (2003)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 181–197 (2002)
8. Fleischer, M.: The measure of Pareto optima. applications to multi-objective metaheuristics. In: Fonseca, C.M., et al. (eds.) EMO 2003. LNCS, vol. 2632, pp. 519–533. Springer, Heidelberg (2003)

[2] Source code is available from `http://iridia.ulb.ac.be/~manuel/archivers` and sequences from `http://iridia.ulb.ac.be/~manuel/archivers-sequences`

9. Hanne, T.: On the convergence of multiobjective evolutionary algorithms. European Journal of Operational Research 117(3), 553–564 (1999)
10. Hansen, M.P.: Metaheuristics for multiple objective combinatorial optimization. Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark (1998)
11. Knowles, J.D.: Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization. Ph.D. thesis, University of Reading, UK (2002)
12. Knowles, J.D., Corne, D.: On metrics for comparing non-dominated sets. In: Proceedings of the 2002 Congress on Evolutionary Computation Conference (CEC 2002), pp. 711–716. IEEE Press, Piscataway (2002)
13. Knowles, J.D., Corne, D.: Properties of an adaptive archiving algorithm for storing nondominated vectors. IEEE Transactions on Evolutionary Computation 7(2), 100–116 (2003)
14. Knowles, J.D., Corne, D.: Bounded Pareto archiving: Theory and practice. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds.) Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, pp. 39–64. Springer, Berlin (2004)
15. Laumanns, M.: Stochastic convergence of random search to fixed size Pareto set approximations. Arxiv preprint arXiv:0711.2949 (2007)
16. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evolutionary Computation 10(3), 263–282 (2002)
17. Rudolph, G., Agapie, A.: Convergence properties of some multi-objective evolutionary algorithms. In: Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000), vol. 2, pp. 1010–1016. IEEE Press, Piscataway (2000)
18. Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary computation and convergence to a Pareto front. In: Koza, J.R. (ed.) Late Breaking Papers at the Genetic Programming 1998 Conference, pp. 221–228. Stanford University Bookstore, Stanford University (1998)
19. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K., et al. (eds.) Evolutionary Methods for Design, Optimisation and Control, pp. 95–100. CIMNE, Barcelona (2002)
20. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., et al. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
21. Zitzler, E., Thiele, L., Bader, J.: SPAM: Set preference algorithm for multiobjective optimization. In: Rudolph, G., et al. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 847–858. Springer, Heidelberg (2008)
22. 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)

On a Stochastic Differential Equation Approach for Multiobjective Optimization up to Pareto-Criticality

Ricardo H.C. Takahashi[1], Eduardo G. Carrano[2], and Elizabeth F. Wanner[2]

[1] Universidade Federal de Minas Gerais, Department of Mathematics
Av. Antonio Carlos, 6627, Belo Horizonte, MG, 31270-901, Brazil
taka@mat.ufmg.br
[2] Centro Federal de Educação Tecnológica de Minas Gerais,
Department of Computer Engineering Av. Amazonas, 7675, Belo Horizonte, MG,
30510-000, Brazil
egcarrano@dppg.cefetmg.br, efwanner@dppg.cefetmg.br

Abstract. Traditional Evolutionary Multiobjective Optimization techniques, based on derivative-free dominance-based search, allowed the construction of efficient algorithms that work on rather arbitrary functions, leading to Pareto-set sample estimates obtained in a single algorithm run, covering large portions of the Pareto-set. However, these solutions hardly reach the exact Pareto-set, which means that Pareto-optimality conditions do not hold on them. Also, in problems with high-dimensional objective spaces, the dominance-based search techniques lose their efficiency, up to situations in which no useful solution is found. In this paper, it is shown that both effects have a common geometric structure. A gradient-based descent technique, which relies on the solution of a certain stochastic differential equation, is combined with a multiobjective line-search descent technique, leading to an algorithm that indicates a systematic solution for such problems. This algorithm is intended to serve as a proof of concept, allowing the comparison of the properties of the gradient-search principle with the dominance-search principle. It is shown that the gradient-based principle can be used to find solutions which are truly Pareto-critical, satisfying first-order conditions for Pareto-optimality, even for many-objective problems.

1 Introduction

Since the early days of Evolutionary Multicriterion Optimization (EMO) theory, the main arguments in favor of the employment of EMO techniques instead of classical mathematical programming techniques in multiobjective optimization problems (MOPs) have been stated as:

- Evolutionary algorithms are able to perform a global search, solving MOPs even in the case of multimodal functions, with disconnected Pareto-sets;
- Evolutionary algorithms do not rely on gradient calculations, which means that they may be used even in the case of non-differentiable, discontinuous, or noisy functions;

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 61–75, 2011.

- EMO algorithms are able to produce a set of several Pareto-set sample solutions in a single algorithm run, unlike the classical techniques, which require one algorithm run to find each non-dominated solution;
- EMO algorithms are able to generate automatically a set of well-distributed samples of the Pareto-set, in contrast with the classical techniques, that require the variation of the algorithm parameters in order to perform a sweep of the Pareto-set.

Those reasons indeed provided motivation for the development of EMO techniques by almost 20 years. The EMO paradigm is usually articulated, by contrast with the classical nonlinear programming paradigm, according to the (somewhat loose) association described as:

Paradigm	**Problem functions**	**Method features**
Classical	unimodal, differentiable	deterministic, gradient-based
EMO	multimodal, non-differentiable	stochastic, gradient-free

This paper revisits the foundations of such a paradigm delimitation, raising back the possibility of building algorithms that: (i) deal with multimodal and differentiable functions; and (ii) employ the principle of stochastic search, using a gradient-based machinery. The motivations for this study may be stated by the following reasoning:

- Although the derivative-free dominance-based search principle of most evolutionary algorithms indeed allows to perform the optimization of rather generic functions under the only assumption of *weak locality*, there is still a large class of functions of practical relevance, for which the derivative information is available.
- When a function belongs to a class for which some gradient-based descent procedure may be applied, the usage of such a kind of procedure is often the most efficient way to perform its optimization, frequently leading to search procedures that are orders of magnitude faster than other procedures such as cutting planes, branch-and-bound, genetic algorithms, and others.
- The search direction methods are usually associated to deterministic searches, which are inconvenient for the purpose of finding a representative set of samples of Pareto-sets. On the other hand, evolutionary methods employ stochastic searches that make the algorithm to visit the “promising” portions of the decision variable space, while a selection procedure provides a dynamic equilibrium which leads the Pareto-set to become a kind of invariant set that contains the algorithm stable fixed-points. As a result, some neighborhood of all solutions in the Pareto-set get a non-null probability of being visited.
- A meaningful combination of stochastic motion and descent-based search might be a basis for the construction of algorithms which are endowed with the computational efficiency of descent algorithms, and also with the ability of sampling the continuous sets of interest, which are to become stable equilibrium sets of the algorithm.

– As a by-product of the descent-based search, it allows to reintroduce gradient-based analytical conditions for Pareto-criticality (the Kuhn-Tucker condition for efficiency).

The discussion presented here has connections with some recent studies concerning: (i) the hybridization of EMO techniques with deterministic nonlinear programming techniques, for performing local searches [4,10,14]; and (ii) the investigation of the nature of the loss of efficiency of dominance-based search techniques in many-objective problems [6,1,9]. The analysis provided by [9] elucidated the mechanism of loss of efficiency of dominance-based search algorithms in many-objective problems as a matter related to the relative size of the cone of dominated solutions, compared to its complement. That reference did not mention the closely related fact that the same phenomenon of shrinking of the dominated cones also causes the loss of efficiency of the dominance-based mechanisms nearby the Pareto-sets, even in problems with few objectives. This effect can be understood if one realizes that the cone of dominated solutions is bounded by the planes which are normal to the objective function gradient vectors, and this cone exactly degenerates to zero volume when the reference point becomes Pareto-optimal, which is equivalent to the Kuhn-Tucker condition for efficiency being attained at that point. Therefore, instead of having two different effects causing difficulties in many-objective problems and in reaching Pareto-optimality in problems with few objectives, it should be recognized that these are the same problem. The adoption of gradient-search strategies is perhaps the only structural solution for this kind of problem, in the sense that the only geometrical entity that can be accommodated safely inside an arbitrarily shrunk cone is a line – which suggests a gradient search.

The inspiration for the study presented here can be found in some works which are contemporary of the first papers on the theme of EMO. In 1994, Ritter and Schaffler proposed an approach for globally solving (single-objective) optimization problems on multimodal functions, using a method that relied on the simulation of a stochastic differential equation [7]. A stochastic dynamic system with Itô structure, whose equilibrium points were the optimization problem optima, was defined. The evolution of this system was simulated using a new predictor-corrector method. The trajectories of this system essentially followed the direction opposite to the gradient vector of the objective function. This dynamic system was shown to visit the several system minima, with the stochastic part of the dynamics providing the jumps from one attraction basin to another one. This approach combined the paradigm of "gradient descent", which was typical of classical approaches, with the stochastic search, that was central in the evolutionary computation approach, in order to perform a global search of the optima of differentiable functions with complex landscapes.

In 2002, Schaffler, Schultz and Weinzierl extended that approach to the multiobjective case [8] (the algorithm presented in that paper will be called here the *SSW algorithm*, for brevity) . A descent direction, calculated using the gradients of the objective functions, and indicating a direction in which there are points which dominate the current one, was used in that work in order to define the dynamic system trajectories. In this way, the first-order Pareto-critical points become the equilibrium points of the deterministic part of the stochastic dynamic system. In that paper, it has been shown that any arbitrarily small neighborhood of any Pareto-critical point is visited, almost surely, by the dynamical system. Although that work has not reached yet a significant impact in the research on multiobjective optimization, it raises the possibility of performing a meaningful combination of stochastic search methods with gradient descent methods, in order to perform the task of Pareto-set estimation.

In 2007, Shukla and Deb evaluated the performance of SSW algorithm, in a comparison with with the classical EMO algorithm NSGA-II and other algorithms, in benchmark problems with up to three objectives [11]. That study shown that the SSW algorithm presented two main drawbacks: (i) It shown severe difficulties for finding well-distributed Pareto-optimal solutions; and (ii) When the search point becomes near the Pareto-set, the stochastic component of the search becomes dominant. This is the main mechanism that makes the search point to visit the whole Pareto-set, but this also generates a drift behavior, which leads the search point to become within a neighborhood of the Pareto-set, without ever reaching exact solutions from this set.

However, the same study [11] concluded that the gradient-based search of SSW algorithm was able to generate solution sets quickly, which indicated that further studies concerning its hybridization with EMO techniques should be conducted. Some numerical experiments presented here suggest that the SSW algorithm has an interesting property that was not focused in [11]: it is able to deal with problems with high number of objective functions. When this number grows, the SSW algorithm becomes increasingly better than dominance-based algorithms, up to situations in which only SSW algorithm works.

This paper presents a hybridization of the SSW algorithm, with the specific intent to solve drawback (ii) mentioned above. A two-step search is employed, using SSW algorithm as the first step. The non-dominated points resulting from SSW are introduced in a local search procedure, which is based on a new multiobjective golden section line search procedure, proposed in [13]. Such a local search machinery is particularly useful in this case, because it generates Pareto-critical points which necessarily dominate the initial solution (this means that if the initial solution set is well distributed, the final solution set is likely to keep this feature), and it remains efficient even for high dimension problems. In this way, a set of exact Pareto-critical solutions (which verify the Kuhn-Tucker conditions for efficiency) is found, with a small computational overhead in relation to the basic SSW.

The results of preliminary tests suggest that the proposed direction of research is promising for dealing, in a unifying way, with the following challenges in continuous-variable problems: (i) many-objective problems, and (ii) finding exact Pareto-critical points. The problem of generation of a homogeneous sampling of the Pareto-set is left for future research, possibly following the guidelines presented in [12].

1.1 Notation

The following operators, applied to vector arguments, mean:

$(\cdot \leq \cdot)$ Each coordinate of the first argument is less than or equal to the corresponding coordinate of the second argument;
$(\cdot < \cdot)$ Each coordinate of the first argument is smaller than the corresponding coordinate of the second argument;
$(\cdot \prec \cdot)$ Each coordinate of the first argument is less than or equal to the corresponding coordinate of the second argument, and at least one coordinate of the first argument is strictly smaller than the corresponding coordinate of the second argument. This relation is read as: the first argument *dominates* the second one.

The operators $(\geq)$, $(>)$ and $(\succ)$ are defined in the analogous way.

2 Preliminary Definitions and Problem Statement

Consider the *multiobjective optimization problem* (MOP) defined by the minimization (w.r.t. the partial order $\leq$) of a vector of objective functions $F(x) = (F_1(x), F_2(x), ..., F_m(x))$:

$$\begin{gathered} \min \; F(x) \\ \text{subject to} \;\; x \in \Omega \end{gathered} \tag{1}$$

where $F_i(x) : \mathbb{R}^n \mapsto \mathbb{R}$ are differentiable functions, for $i = 1, ..., m$, and $\Omega \subset \mathbb{R}^n$ is the feasible set, defined by

$$\Omega \triangleq \{x \in \mathbb{R}^n \mid g(x) \leq 0\}\,, \tag{2}$$

with $g(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^p$ a vector of differentiable functions. Associated to the minimization of $F(\cdot)$, the *efficient solution set*, Ω^*, is defined as:

$$\Omega^* \triangleq \{x^* \in \Omega \mid \;\nexists\, x \in \Omega \text{ such that } F(x) \prec F(x^*)\} \tag{3}$$

The *multiobjective optimization problem* is defined as the problem of finding vectors $x^* \in \Omega^*$. This set of solutions is also called the *Pareto-set* of the problem.

This paper is concerned with the problem of finding vectors which satisfy certain conditions for belonging to Ω^*. The following matrices are defined:

$$\begin{cases} H(x) = \left[\nabla F_1(x)\ \nabla F_2(x) \ldots \nabla F_m(x)\right] \\ \\ G(x) = \left[\nabla g_{\mathcal{J}(1)}(x)\ \nabla g_{\mathcal{J}(2)}(x) \ldots \nabla g_{\mathcal{J}(r)}(x)\right] \\ \\ W(x) = \left[H(x)\ G(x)\right] \end{cases} \tag{4}$$

in which $\mathcal{J}$ denotes the set of indices of the active constraints, with r elements. Then, $g_i(x) = 0 \Leftrightarrow i \in \mathcal{J}$.

The *linearized feasible cone* at a point x, denoted by $\mathcal{G}(x)$, is defined as:

$$\mathcal{G}(x) \triangleq \{\omega \in \mathbb{R}^n \mid G'(x) \cdot \omega \leq 0\} \tag{5}$$

Given $x \in \Omega$, a vector $\omega \in \mathbb{R}^n$ is a *tangent direction* of Ω at x if there exists a sequence $[x_k]_k \subset \Omega$ and a scalar $\eta > 0$ such that:

$$\lim_{k\to\infty} x_k = x \ , \qquad \text{and} \qquad \lim_{k\to\infty} \eta \frac{x_k - x}{\|x_k - x\|} = \omega \tag{6}$$

The set of all tangent directions is called the *contingent cone* of Ω at x, and is denoted by $\mathcal{T}(\Omega, x)$. In this paper, the following constraint qualification (see reference [5]) is assumed to hold:

$$\mathcal{T}(\Omega, x) = \mathcal{G}(x) \tag{7}$$

Theorem 1. *Consider the multiobjective optimization problem defined by (1) and (2), and assume that the constraint qualification (7) holds. Under such assumption, a necessary condition for $x^* \in \Omega^*$ is that there exist vectors $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^r$, with $\lambda \succ 0$ and $\mu \geq 0$, such that:*

$$H(x^*) \cdot \lambda + G(x^*) \cdot \mu = 0 \tag{8}$$

■

This theorem is a matrix formulation of the Kuhn-Tucker necessary conditions for efficiency (KTE), that become also sufficient in the case of convex problems (see, for instance, [2]). The points x^* which satisfy the conditions of Theorem 1 are said to be *first-order Pareto-critical points*. This paper is concerned with the search for such points.

3 Dominating Cones

The structural difficulties that are inherent to dominance-based search algorithms are discussed in this section, with the aid of two "prototypical" problems, one for the *Pareto-set proximity effect* and the other one for the *objective space dimension effect*.

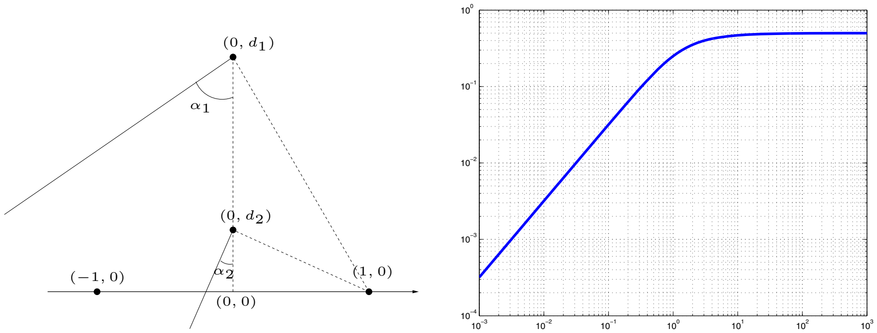
Pareto-proximity effect

Consider a bi-objective problem in a two-variable space, with the objective functions:

$$\begin{gathered} f_1(x) = (x - x_1)'(x - x_1) \\ f_2(x) = (x - x_2)'(x - x_2) \\ x_1 = \begin{bmatrix} 1 \ 0 \end{bmatrix}' \quad x_2 = \begin{bmatrix} -1 \ 0 \end{bmatrix}' \end{gathered} \tag{9}$$

Consider now a point at coordinates $x_a = [0 \ d]'$. The cone with vertex in x_a that contains the points that dominate x_a has internal angle 2α, with alpha given by:

$$\alpha = \frac{\pi}{2} - \arctan\left(\frac{1}{d}\right) \tag{10}$$



(a) (b)

Fig. 1. (a) The cone of points that dominate $(0, d_1)$ makes an angle of α_1 in relation to the vertical, while the cone of points which dominate $(0, d_2)$ makes an angle α_2 in relation to the vertical. (b) Representation of $p_\alpha(d)$, with d represented in the horizontal axis and p_α represented in the vertical axis.

This relation can be drawn by direct inspection of figure 1(a). This means that a mutation applied on x_a with uniform probability of falling in any direction would have a probability $p_\alpha(d)$ of generating another point inside that cone:

$$p_\alpha(d) = 1 - \frac{2}{\pi} \arctan\left(\frac{1}{d}\right) \tag{11}$$

The graphics of $p_\alpha(d)$ is represented in figure 1(b), in logarithmic coordinates. This graphics reveals the following pattern: (i) for large d, when the point x_a is distant from both x_1 and x_2, the probability of generating a mutation inside the dominating cone remains nearby 0.5. There is a qualitative change in this behavior when x_a reaches a distance that is of the order of the distance between the two minima of individual functions, x_1 and x_2. After reaching this distance, the probability of further enhancements in x_a start to decrease at the same rate of the decreasing of distance d. This means that each time the distance d is reduced to a half of a former value, the probability of a further enhancement becomes divided by two too.

Clearly, there will be a distance d in which the probability of finding any point that dominates x_a becomes very small, even for a large number of independent attempts. This phenomenon should be considered as the structural reason behind the largely known incapacity of dominance-based algorithms to reach Pareto-critical points. These algorithms stagnate at some distance from the Pareto-optimal set, and the attainment of points belonging to the set is possible only via local search procedures.

Objective space dimension effect

Another effect that is very similar in structure to the Pareto-proximity effect is related to the growth in the dimension of objective space. Consider now an n-objective problem, in a decision variable space also with n dimensions:

$$\begin{aligned} & f_1(x) = (x - x_1)'(x - x_1) \\ & f_2(x) = (x - x_2)'(x - x_2) \\ & \cdots \\ & f_n(x) = (x - x_n)'(x - x_n) \\ & x_1 = \left[1\; 0 \cdots 0 \right]' \;\; \ldots \;\; x_n = \left[0 \cdots 0\; 1 \right]' \end{aligned} \tag{12}$$

Consider now the point $x_a = (0, 0, \ldots, 0)$, situated at the origin of the coordinate system. Clearly, the orthant of positive coordinates includes all points that dominate x_a, while the other regions of space have points that do not dominate x_a. The probability of a mutation with random direction with uniform probability will present the following probability $p_\beta(n)$ of falling into the positive orthant:

$$p_\beta(n) = \frac{1}{2^n} \tag{13}$$

Equation 13 shows that the probability of a random mutation to fall into the positive orthant is divided by 2^k when the dimension of the objective space is multiplied by k. This is the structural cause of the difficulty of dominance-based algorithms when dealing with the so-called many-objective problems. It should be noticed that the objective space dimension effect is very similar to the Pareto-proximity effect, in the sense that both effects, beyond some threshold, necessarily lead to a stagnation behavior or dominance-based search mechanisms. It also should be noticed that both effects compose in the same problem in a multiplicative way. Finally, it is worthy to point out that both effects can be avoided if a search is conducted toward the dominating cone, for instance using gradient information.

4 The SSW Algorithm

The SSW algorithm, as presented in [8], is concerned with unconstrained multiobjective optimization problems (1), in which the feasible set $\Omega \equiv \mathbb{R}^n$. This section describes this algorithm, following closely the development presented in [8]. All theorems and the resulting algorithm in this section come from that reference.

Consider the following quadratic optimization problem for each $x \in \mathbb{R}^n$:

$$\min_{\alpha \in \mathbb{R}^m} \left\{ \left\| \sum_{i=1}^{m} \alpha_i \nabla f_i(x) \right\|^2 , \; \alpha_i \geq 0 \, , \; i = 1, \ldots, m \, , \; \sum_{i=1}^{m} \alpha_i = 1 \right\} , \tag{14}$$

where $\nabla f_i(x)$ denotes the gradient of the i-th component of the objective function vector. Let $\hat{\alpha}$ be a global minimizer of (14), and define the following function:

$$q : \mathbb{R}^n \to \mathbb{R}^n \ , \ x \mapsto \sum_{i=1}^{m} \hat{\alpha}_i \nabla f_i(x), \tag{15}$$

It can be shown that either $q(x) = 0$ or $-q(x)$ is a descent direction for all functions $f_1, \ldots, f_m$ at x; hence, each x with $q(x) = 0$ is Pareto-critical. The properties of function $q(x)$ allow to define the following initial value problem (IVP) for multiobjective optimization:

$$\dot{x}(t) = -q(x(t)) \tag{16}$$

The following Theorem holds.

Theorem 2. *Let $x_0 = x(0)$. Consider problem (1) and the corresponding IVP (16), with $q(x_0) \neq 0$. Assume that the set of points which dominate x_0 is bounded for all $x_0 \in \mathbb{R}^n$. Then, a solution $x(t) : [0, \infty[\to \mathbb{R}^n$ of (16) exists and is unique for all $x_0 \in \mathbb{R}^n$, with*

$$f(t) \prec f(t_1)\,,\, \forall\, 0 \leq t_1 < t < \infty.$$

■

As a consequence of Theorem 2, a numerical resolution of (16) leads to a single Pareto-critical solution. The following class of Itô stochastic differential equations is proposed, in order to generate a large set of Pareto-critical solutions:

$$dX_t = -q(X_t)dt + \epsilon dB_t, \tag{17}$$

where $\{B_t\}$ is an n-dimensional Brownian motion, $q(\cdot)$ is defined by (15), $\epsilon > 0$ and $x_0 \in \mathbb{R}^n$.

Algorithm 1. - SSW Algorithm -

Input: $q(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $\epsilon > 0$, $\sigma_0 > 0$, j_{max}

Output: x_t.

```
$\sigma \leftarrow \sigma_0$
while $j < j_{max}$ do
    $n_1 \leftarrow N(0, I_n)$
    $n_2 \leftarrow N(0, I_n)$
    $n_3 \leftarrow n_1 + n_2$
    $x^1_{j+1} \leftarrow x_j - \sigma q(x_j) - \epsilon n_3 (\sigma/2)^{\frac{1}{2}}$
    $x(\sigma/2) = x_j (\sigma/2) q(x_j) - \epsilon n_1 (\sigma/2)^{\frac{1}{2}}$
    $x^2_{j+1} \leftarrow x(\sigma/2) - (\sigma/2) q(x(\sigma/2)) - \epsilon n_2 (\sigma/2)^{\frac{1}{2}}$
    if $\|x^1_{j+1} - x^2_{j+1}\|_2 \leq \delta$ then
        $x_{j+1} \leftarrow x^2_{j+1}$
    else
        $\sigma \leftarrow \sigma/2$
    end if
end while
```

In [8] it is shown that, under some technical hypothesis for the function vector $F(\cdot)$, for each Pareto-critical solution $\bar{x}$, almost all paths of $\{X_t\}$ hit any ball $S(\bar{x}, \rho)$ centered at $\bar{x}$ for any arbitrarily chosen $\rho > 0$, after a finite time, for all $x_0 \in \mathbb{R}^n$. For the numerical computation of a path of $\{X_t\}$, the iterative scheme of Algorithm 1 is used.

5 The SSW-LS Algorithm

The lack of ability of the basic SSW algorithm to find Pareto-critical points, despite the usage of gradient information, motivates the algorithm hybridization presented here, the SSW-LS algorithm (SSW with Local Search algorithm). The idea is simply to take the non-dominated points, from the collection of points generated by SSW algorithm, and run a gradient-based local search procedure starting in each one, finishing in Pareto-critical points.

Some classical gradient-based iterative methods for multiobjective optimization would present difficulties for being inserted in the structure of SSW-LS. For instance, weighted-sum procedures would not search for Pareto-critical points which are near to initial solutions [2]. Instead, this method would lead to specific Pareto-critical points in the Pareto-set, not necessarily near to the initial point. The references [4,10] employed an adaptation of the ϵ-constrained search, while [14] employed an adaptation of the goal attainment procedure, for performing local searches in multiobjective evolutionary algorithms. However, those alternatives, based on scalarizations of the multiobjective problem, do not take advantage of the multiobjective problem structure, relying on mono-objective algorithms. One should notice that each time a scalarized algorithm runs, it searches for a specific Pareto-optimal point that is implicitly defined as the minimum of the scalarized objective function. However, the evolutionary algorithm is searching for a set of representative points of the Pareto-set, without an specific interest for any point. This means that some computational effort is spent in order to identify points which are not *per se* relevant, while a much smaller effort could have been spent if the search were directed toward any efficient point that could be find.

A multiobjective descent line-search procedure, based on a new multiobjective golden section unidimensional search, has been proposed in [13]. This procedure fits the need of SSW-LS algorithm, using the descent direction given by vector $-q(x)$, and an efficient multiobjective line search algorithm that contracts the search segment faster than the traditional mono-objective golden section procedure. As the algorithm is not targeted to any specific point, it can perform faster searches that identify Pareto-critical points. In [13], it is shown that this procedure remains efficient even for high dimension problems, with hundreds of variables. A brief description of the line search procedure is presented in the next subsection.

The Local Search procedure of algorithm SSW-LS is presented in Algorithm 2. The input set $\{\bar{x}_1, \ldots, \bar{x}_p\}$ comes from the SSW algorithm, and includes only the non-dominated points of the output of SSW.

Algorithm 2. - LS Procedure -

Input: $q(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, $\{\bar{x}_1, \ldots, \bar{x}_p\} \subset \mathbb{R}^n$, $\epsilon_q > 0$

Output: $\{x_1^*, \ldots, x_p^*\}$ % a set of Pareto-critical points

```
for $i \leftarrow 1 : p$ do
    $x \leftarrow \bar{x}_i$
    while $\|q(x)\|_2 \geq \epsilon_q$ do
        $d \leftarrow -q(x)$
        $x \leftarrow$ multiobjective_golden_section$(x, d, F)$
    end while
    $x_i^* \leftarrow x$
end for
```

5.1 The Multiobjective Golden Section Line Search

The detailed proof of the statements presented in this section can be found in [13]. The idea is to find a point x^* which dominates the current point x_0, such that x^* belongs to the Pareto-set of problem (1) constrained to the search direction. The information available to be used in a line search procedure is the set of values of function $F(\cdot)$ in points $\alpha = 0$ (which represents initially the current point), $\alpha = \alpha_A$ and $\alpha = \alpha_B$. The variable α parametrizes the line segment in which the search is conducted. For brevity, denote: $f_0 = F(0)$, $f_A = F(\alpha_A)$ and $f_B = F(\alpha_B)$. The vector function $C(\cdot,\cdot,\cdot) : \mathbb{R}^{m\times 3} \mapsto \{0,1\}^6$ is defined as:

$$\begin{aligned} C(f_0, f_A, f_B) &= \left[\, C_1(f_0, f_A, f_B) \ldots C_6(f_0, f_A, f_B) \,\right] = \\ &= \left[\, (f_0 \succ f_A)\ (f_0 \succ f_B)\ (f_A \succ f_B)\ (f_0 \prec f_A)\ (f_0 \prec f_B)\ (f_A \prec f_B) \,\right] \end{aligned} \tag{18}$$

with each C_i a binary number, 1 meaning "`true`" and 0 meaning "`false`" for the result of each comparison. For instance $f_0 \succ f_A$ makes $C_1 = 1$, otherwise, $C_1 = 0$ ($f_0 \not\succ f_A$). There are three possible operations for contracting the current trust-segment (i.e., the segment in which it is known that there exists some point that belongs to the line-constrained Pareto-set) in a line search that is based on the function evaluation on points $\alpha = 0$, $\alpha = \alpha_A$ and $\alpha = \alpha_B$. Let a current trust-segment be $[\alpha_1, \alpha_2] \subset [0, 1]$. These operations are named D1, D2 and D3, as indicated in Table 1.

A decision table that "turns on" each contraction operation is shown in Table 2. This decision table maximizes the contraction of the trust-segment, without allowing the loss of the line-constrained Pareto set [13].

Table 1. The contraction operations on the trust segment $[\alpha_1, \alpha_2] \subset [0, 1]$

	Contraction operation
D1	discard $[0, \alpha_A] \cap [\alpha_1, \alpha_2]$
D2	discard $[\alpha_A, \alpha_B] \cap [\alpha_1, \alpha_2]$
D3	discard $[\alpha_B, 1] \cap [\alpha_1, \alpha_2]$

Table 2. Operations performing the maximal trust region contraction

Condition	Operation
$C_1 \cdot \overline{C_6}$	D1
$\overline{C_1}$	D2
$C_1 \cdot C_2 \cdot C_3 \cdot \overline{C_4} \cdot \overline{C_5} \cdot \overline{C_6}$	D3

The *Golden Section Multiobjective Line Search* algorithm implements the segment contraction procedures using the golden section ratio, leading to a line-constrained Pareto-optimal solution which necessarily dominates the initial point.

5.2 Results of SSW-LS Algorithm

The following vector function, which can be constructed for any number n of decision variables and for any number m of objectives (with $m \leq n$), has been used here:

$$F_i(x) = (x - x_i)' A_i (x - x_i) \tag{19}$$

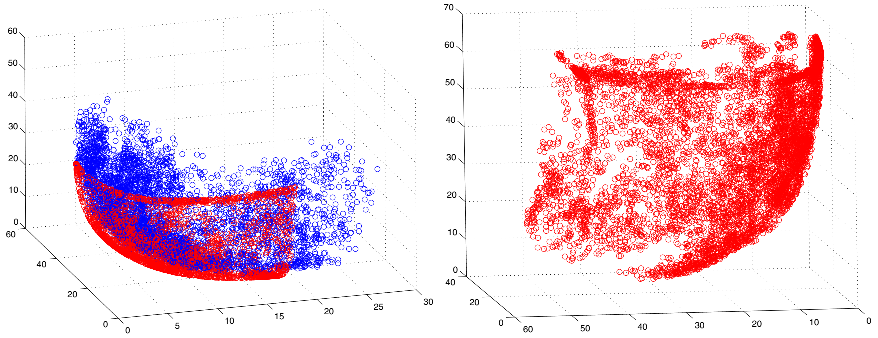
in which:

$$A_i(j,k) = \begin{cases} 0\,, & \text{if } j \neq k \\ 1\,, & \text{if } j = k \text{ and } j \neq i \\ i\,, & \text{if } j = k = i \end{cases} \quad , \qquad x_i(j) = \begin{cases} 0\,, & \text{if } j \neq i \\ m\,, & \text{if } j = i \end{cases}$$

The results of some experiments with SSW-LS algorithm are shown in Figure 2. These experiments were run with $\epsilon = 0.3$, and each gradient evaluation is counted as $(n + 1)$ function evaluations.

Figure 2(a) presents a comparison of the Pareto-fronts (the results in the objective space) obtained with the basic SSW algorithm and with the SSW-LS algorithm, in the problem instance with 10 decision variables and 3 objective functions, and with 10000 function evaluations assigned to each algorithm. It becomes apparent that the SSW-LS algorithm provides a much more well-defined estimation of the Pareto-set, with solutions that dominate the basic SSW ones, and with a clear definition of the Pareto-front boundaries.

Figure 2(b) deals with the problem instance with 10 decision variables and 4 objectives. This figure presents the projection, in the space of the first 3 objectives, of the Pareto-front obtained with SSW-LS. This entity is a solid object in the $\mathbb{R}^3$ space, bounded by 2-dimensional surfaces. It should be noticed that the solution set delivered by SSW-LS provides a good description of the edges and of the facets of this object.



(a)

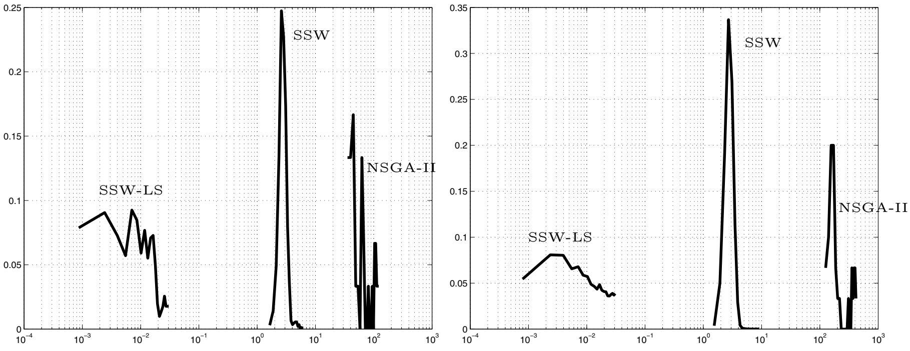
(b)

Fig. 2. (a) Comparison of SSW solutions and SSW-LS solutions, for $n = 10$ and $m = 3$. The SSW solutions delineate a more regular geometric object, with well-defined edges. (b) SSW-LS solutions, projected in the space of the first 3 objectives, for $n = 10$ and $m = 4$.

6 Results in Many-Objective Problems

An experiment has been conducted for the same function described in (19), now with 30 decision variables and 10 objective functions. Three algorithms have been compared: (i) a Matlab version of the canonical NSGA-II [3]; (ii) the SSW algorithm; and (iii) the SSW-LS algorithm. As a merit figure, for the analysis of the solution quality, it is used here the norm of $q(x)$. This is reasonable, since $\|q(x)\|_2$ decreases as x approaches the Pareto-set. It is also known that for any Pareto-critical point, $\|q(x)\|_2 = 0$, which means that this figure also represents an absolute criterion that indicates if a point features Pareto-criticality. Figure 3 shows the results of this experiment.

Figure 3 shows the histograms of the value of $\|q(x)\|_2$ for the solution sets obtained by the three algorithms, for a run with 80000 function evaluations, and for a run with 1×10^6 function evaluations. The similarity of the histograms for such different numbers of function evaluations suggest that a kind of "terminal set" has been reached, for all algorithms, within 80000 function evaluations. Both the NSGA-II and SSW algorithms seem to be unable to further increase the quality of their solution sets. It should be noticed that, as expected, the NSGA-II has found a set of points that is very far from the Pareto-set, featuring $\|q(x)\|_2 \approx 100$. Also as expected, the SSW algorithm has found points that are located near to the Pareto-set, with $\|q(x)\|_2 \approx 1$. Further enhancements in the solution set, in this case, seem to be difficult due to the dominance of the stochastic term in the stochastic differential equation. On the other hand, the SSW-LS algorithm has reached the exact Pareto-set in both cases, within the numerical precision that has been established for these experiments ($\|q(x)\|_2 \approx \epsilon_q = 1 \times 10^{-2}$).



(a) (b)

Fig. 3. Comparison of SSW solutions, SSW-LS solutions and NSGA-II solutions, for $n = 30$ and $m = 10$. The graphics represent the histogram, in logarithm scale, of the value of $\|q(x)\|_2$ for the solution set of each algorithm, for the cases of: (a) 80000 function evaluations, (b) 1×10^6 function evaluations.

7 Conclusions

This paper presented an algorithm for multiobjective optimization, the SSW-LS algorithm, based on the stochastic differential equation approach of [8] and on the multiobjective line search procedure of [13], which is intended to develop some analysis concerning the structural causes of seemingly uncorrelated difficulties of multiobjective optimization: (i) many-objective problems; and (ii) problems in which truly Pareto-critical solutions are required. The geometric structure of performing searches toward descent cones seems to underly those problems, and indicates that gradient-based (or other directional-based) searches might constitute a solution. The proposed algorithm is an instance of a more fundamental discussion about the roles of different mechanisms in multiobjective optimization. In synthesis, the current data acquired up to now indicates that (i) to reach the Pareto-set, deterministic steps should be the dominant effect in the algorithm; (ii) to sample the Pareto-set, a stochastic search over an equilibrium set should play a main role; and (iii) to reach exact Pareto-critical solutions, deterministic procedures performing a descent search, finding solutions that dominate the initial point, should be employed.

References

1. Adra, S.F., Fleming, P.J.: Diversity management in evolutionary many-objective optimization. IEEE Transactions on Evolutionary Computation (to appear, 2011)
2. Chankong, V., Haimes, Y.Y.: Multiobjective Decision Making: Theory and Methodology. Elsevier, Amsterdam (1983)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA II. IEEE Trans. Evol. Comp. 6(2), 182–197 (2002)

4. Deb, K., Tewari, R., Dixit, M., Dutta, J.: Finding trade-off solutions close to KKT points using evolutionary multi-objective optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, pp. 2109–2116 (2007)
5. Gould, F.J., Tolle, J.W.: A necessary and sufficient qualification for constrained optimization. SIAM J. Appl. Math. 20(2), 164–172 (1971)
6. Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. IEEE Transactions on Evolutionary Computation 11(6), 770–784 (2007)
7. Ritter, K., Schaffler, S.: A stochastic method for constrained global optimization. SIAM J. Optim. 4(4), 894–904 (1994)
8. Schaffler, S., Schultz, R., Weinzierl, K.: Stochastic method for the solution of unconstrained vector optimization problems. J. Optim. Theor. Appl. 114(1), 209–222 (2002)
9. Schütze, O., Lara, A., Coello, C.A.C.: On the influence of the number of objectives on the hardness of a multiobjective optimization problem. IEEE Transactions on Evolutionary Computation (to appear, 2011)
10. Sharma, D., Kumar, A., Deb, K., Sindhya, K.: Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, pp. 3003–3010 (2007)
11. Shukla, P.K., Deb, K.: On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. European Journal of Operational Research 181, 1630–1652 (2007)
12. Takahashi, R.H.C., Guimaraes, F.G., Wanner, E.F., Carrano, E.G.: Feedback-control operators for evolutionary multiobjective optimization. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 66–80. Springer, Heidelberg (2009)
13. Vieira, D.A.G., Takahashi, R.H.C., Saldanha, R.R.: Multicriteria optimization with a multiobjective golden section line search. Mathematical Programming (to appear, 2011) doi: 10.1007/s10107-010-0347-9
14. Wanner, E.F., Guimaraes, F.G., Takahashi, R.H.C., Fleming, P.J.: Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. Evolutionary Computation 16(2), 185–224 (2008)

Pareto Cone ϵ-Dominance: Improving Convergence and Diversity in Multiobjective Evolutionary Algorithms

Lucas S. Batista, Felipe Campelo, Frederico G. Guimarães,
and Jaime A. Ramírez*

Universidade Federal de Minas Gerais, Departamento de Engenharia Elétrica
Av. Antônio Carlos 6627, Belo Horizonte 31720-010, MG, Brazil
jramirez@ufmg.br

Abstract. Relaxed forms of Pareto dominance have been shown to be the most effective way in which evolutionary algorithms can progress towards the Pareto-optimal front with a widely spread distribution of solutions. A popular concept is the ϵ-dominance technique, which has been employed as an archive update strategy in some multiobjective evolutionary algorithms. In spite of the great usefulness of the ϵ-dominance concept, there are still difficulties in computing an appropriate value of ϵ that provides the desirable number of nondominated points. Additionally, several viable solutions may be lost depending on the hypergrid adopted, impacting the convergence and the diversity of the estimate set. We propose the concept of cone ϵ-dominance, which is a variant of the ϵ-dominance, to overcome these limitations. Cone ϵ-dominance maintains the good convergence properties of ϵ-dominance, provides a better control over the resolution of the estimated Pareto front, and also performs a better spread of solutions along the front. Experimental validation of the proposed cone ϵ-dominance shows a significant improvement in the diversity of solutions over both the regular Pareto-dominance and the ϵ-dominance.

Keywords: Evolutionary multiobjective optimization, evolutionary algorithms, ϵ-dominance, Pareto front.

1 Introduction

The assessment of the quality of estimates of Pareto-optimal fronts produced by evolutionary multiobjective algorithms is itself a multi-criteria problem. A high-quality approximation set should: (i) approach the true Pareto front as close as possible, and (ii) be well-spread along its extension [1]. To fulfill these goals, a Pareto-based fitness assignment method is usually designed in order to guide the search toward the global Pareto-optimal front, whereas density estimation methods as crowding [2] or clustering [3] are commonly employed to preserve the population diversity.

* Corresponding author.

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 76–90, 2011.

A few years ago, Laumanns *et al.* [4] proposed a relaxed form of dominance for MOEAs, called ϵ-dominance. This mechanism acts as an archiving strategy to ensure both properties of convergence towards the Pareto-optimal front and properties of diversity among the solutions found. Since this technique guarantees that no two achieved solutions are within an ϵ_i value from each other in the i^{th} objective, the ϵ value is usually provided by the designer to control the size of the solution set. Nevertheless, as the geometrical features of the Pareto-optimal front are commonly unknown by the designer, the ϵ-dominance strategy can lose a high number of viable solutions when the ϵ value is badly estimated.

In spite of the great usefulness of the ϵ-dominance concept, the way in which the solutions are selected within each hyperbox presents several drawbacks. The main one refers to the difficulties in computing an appropriate value of ϵ to provide the desired number of nondominated points. Moreover, this approach tends to neglect viable solutions located on segments of the Pareto front that are almost parallel to the axes of the objective space, as well as the extreme points of the Pareto front, contributing negatively to the spread of solutions along the extension of the estimated Pareto front [5].

In order to address some of these limitations, we propose a relaxation of the strict dominance concept, based on an extension of the ϵ-dominance scheme named *cone ϵ-dominance*. The cone ϵ-dominance relaxation aims at maintaining the good convergence properties of ϵ-dominance, while providing a better control over the resolution of the estimated Pareto front, providing a dominance criterion that is less sensitive to the geometrical features of the Pareto front than the ϵ-dominance.

This paper is organized as follows: Section 2 reviews the basic definitions of the ϵ-dominance approach. Section 3 contains a detailed description of the cone ϵ-dominance approach. Section 4 describes the performance metrics used to evaluate the proposed strategy, and Section 5 presents a comparative analysis of the proposed approach, and discusses the results obtained. Finally, conclusions and future directions are given in Section 6.

2 Pareto ϵ-Dominance

Although Laumanns *et al.* [4] have proposed two ϵ-dominance methods, only the additive scheme will be discussed hereinafter. Assume that all objectives f_i, $i \in \{1, \ldots, m\}$, are to be minimized, and also that $1 \leq f_i \leq K$, for all i. Then, given a vector $\boldsymbol{y} \in \mathbb{R}^m$ and $\epsilon > 0$, $\boldsymbol{y}$ is said to ϵ-dominate $\boldsymbol{y}' \in \mathbb{R}^m$, denoted as $\boldsymbol{y} \overset{\epsilon}{\prec} \boldsymbol{y}'$, if and only if,

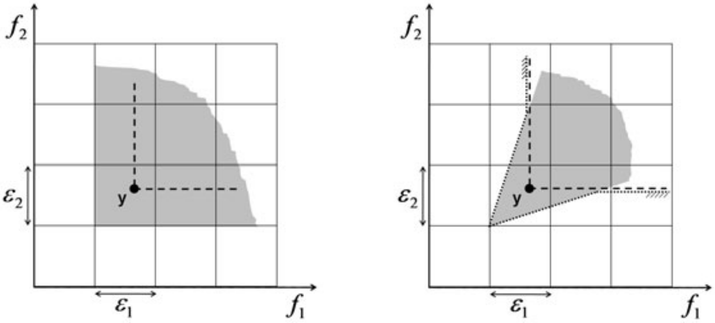
$$y_i - \epsilon \leq y'_i, \text{ for all } i \in \{1, \ldots, m\} \ . \qquad (1)$$

Note that the previous definition can be generalized by considering a different ϵ value for each objective. Essentially, this ϵ-dominance mechanism generates a hypergrid in the objective space with $((K-1)/\epsilon)^m$ boxes which accommodate a maximum of $((K-1)/\epsilon)^{m-1}$ non ϵ-dominated points. Supposing that the designer wants a maximum of T non ϵ-dominated points in the archive, the ϵ value can be easily calculated as[1] $\epsilon = (K-1)/T^{\frac{1}{m-1}}$.

According to Laumanns *et al.* [4], a two-level selection mechanism is implemented in the ϵ-dominance approach. First, it creates a hypergrid in the objective space where each box uniquely contains one vector. Basically, a box-level dominance relation is used, so that the algorithm always maintains a set of nondominated boxes, thus guaranteeing the diversity property. Second, if two vectors share the same box, the usual Pareto dominance relation is applied, so that the best one is selected and convergence is guaranteed. However, if none of these two vectors dominates the other, it is usual to keep the point closest to the origin of the box, i.e., to the corner where all objectives would have the lowest values within that box.

3 Pareto Cone ϵ-Dominance

First, let us present a conceptual interpretation of the proposed cone ϵ-dominance approach. To this end, both ϵ-dominance and cone ϵ-dominance strategies are contrasted in Fig. 1. In order to get a nondominated solution set, the cone ϵ-dominance mechanism entails both the shaded region and the standard Pareto dominance, i.e., the hypervolume dominated by $\boldsymbol{y}$ using the cone ϵ-dominance approach represents a relaxation of that dominated by $\boldsymbol{y}$ when using the usual dominance (see Fig. 1(b)). So, this relaxation enables the approximation of nondominated points in some adjacent boxes that would be ϵ-dominated. Note also that cone ϵ-dominance can be seen as a hybrid between ϵ-dominance and the proper efficiency with respect to cones discussed in [6].



(a) ϵ-dominance.

(b) cone ϵ-dominance.

Fig. 1. Illustration of the ϵ-dominance and cone ϵ-dominance concepts for a bi-objective minimization problem

[1] The ϵ-dominance strategy is only able to obtain this number T in cases where the Pareto-front is linear. For other cases this value is merely an upper limit, with the actual number of nondominated points found being much smaller [5].

3.1 Basic Definitions

Before we present the formal definition of the cone ϵ-dominance strategy, let us define some important concepts.

Definition 1. (Cone) A set $\mathcal{C}$ is a cone if $\lambda \boldsymbol{y} \in C$ for any $\boldsymbol{y} \in C$ and $\forall \lambda \geq 0$. □

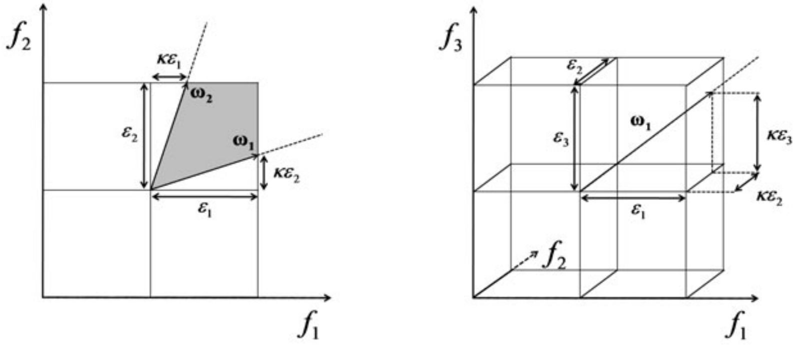
Definition 2. (Generated cone) The cone generated by the vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ is the set $\mathcal{C} = \{\boldsymbol{z} : \boldsymbol{z} = \lambda_1 \boldsymbol{w}_1 + \lambda_2 \boldsymbol{w}_2,\ \forall \lambda_1, \lambda_2 \geq 0\}$. □

Note that the concept of a generated cone can be extended to m dimensions. Thus, the hypercone generated by the vectors $\boldsymbol{w}_i$, $i \in \{1, \ldots, m\}$, is the set $\mathcal{C} = \{\boldsymbol{z} : \boldsymbol{z} = \lambda_1 \boldsymbol{w}_1 + \lambda_2 \boldsymbol{w}_2 + \ldots + \lambda_i \boldsymbol{w}_i + \ldots + \lambda_m \boldsymbol{w}_m,\ \forall \lambda_i \geq 0\}$.

Based on these definitions, we suggest a mechanism to control the hypervolume dominated by a specific cone $\mathcal{C}$. Consider the illustrations in Fig. 2. For the 2D case (Fig. 2(a)), it is easy to see that[2] $\boldsymbol{w}_1 = [\epsilon_1 \ \ \kappa\epsilon_2]^T$ and $\boldsymbol{w}_2 = [\kappa\epsilon_1 \ \ \epsilon_2]^T$. The cone $\mathcal{C}$ can therefore be rewritten as:

$$\mathcal{C} = \left\{ \boldsymbol{z} : \overbrace{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}}^{\boldsymbol{z}} = \overbrace{\begin{bmatrix} \epsilon_1 & \kappa\epsilon_1 \\ \kappa\epsilon_2 & \epsilon_2 \end{bmatrix}}^{\Psi} \overbrace{\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}}^{\boldsymbol{\lambda}},\ \forall \lambda_1, \lambda_2 \geq 0 \right\} \tag{2}$$

in which the parameter $\kappa \in [0, 1)$ controls the opening of the cone $\mathcal{C}$, and Ψ is the cone-dominance matrix, which in fact controls the hypervolume dominated by $\mathcal{C}$. Notice that the cone ϵ-dominance strategy tends toward the ϵ-dominance strategy when $\kappa \to 0$.



(a) Control mechanism in 2D.

(b) Control mechanism in 3D.

Fig. 2. Mechanism used in the cone ϵ-dominance approach to control the hypervolume dominated by a specific cone

[2] With respect to the origin of the box.

For the 3D case (Fig. 2(b)), the process is the same. Though we have plotted only the first vector, it can be seen that $\boldsymbol{w}_1 = [\epsilon_1 \;\; \kappa\epsilon_2 \;\; \kappa\epsilon_3]^T$, $\boldsymbol{w}_2 = [\kappa\epsilon_1 \;\; \epsilon_2 \;\; \kappa\epsilon_3]^T$, and $\boldsymbol{w}_3 = [\kappa\epsilon_1 \;\; \kappa\epsilon_2 \;\; \epsilon_3]^T$. In this case, the cone-dominance matrix is given by:

$$\Psi(\epsilon_1, \epsilon_2, \epsilon_3, \kappa) = \begin{bmatrix} \epsilon_1 & \kappa\epsilon_1 & \kappa\epsilon_1 \\ \kappa\epsilon_2 & \epsilon_2 & \kappa\epsilon_2 \\ \kappa\epsilon_3 & \kappa\epsilon_3 & \epsilon_3 \end{bmatrix} . \tag{3}$$

By induction, when $i \in \{1, \ldots, m\}$, we get $\Psi \mapsto \mathbb{R}^{m \times m}$, i.e.,

$$\Psi(\epsilon_i, \kappa) = \begin{bmatrix} \epsilon_1 & \kappa\epsilon_1 & \ldots & \kappa\epsilon_1 \\ \kappa\epsilon_2 & \epsilon_2 & \ldots & \kappa\epsilon_2 \\ \vdots & \vdots & \ddots & \vdots \\ \kappa\epsilon_m & \kappa\epsilon_m & \ldots & \epsilon_m \end{bmatrix} . \tag{4}$$

Finally, we can formally define the cone ϵ-dominance strategy.

Definition 3. (Cone ϵ-dominance) Given two feasible vectors $\boldsymbol{y}$, $\boldsymbol{y}' \in \mathbb{R}^m$, $\boldsymbol{y}$ is said to cone ϵ-dominate $\boldsymbol{y}'$ if and only if, $\boldsymbol{y}$ Pareto-dominates $\boldsymbol{y}'$ or the solution of the linear system $\Psi\boldsymbol{\lambda} = \boldsymbol{z}$, with $\boldsymbol{z} = \boldsymbol{y}' - [\boldsymbol{y} - \boldsymbol{\epsilon}]$, and $\epsilon_i > 0$, gives $\lambda_i \geq 0 \; \forall i \in \{1, \ldots, m\}$. Equivalently, we say $\boldsymbol{y} \overset{cone\epsilon}{\prec} \boldsymbol{y}'$ if and only if,

$$(\boldsymbol{y} \prec \boldsymbol{y}') \;\; \vee \;\; (\Psi\boldsymbol{\lambda} = \boldsymbol{z} \mid \lambda_i \geq 0 \text{ for all } i = \{1, \ldots, m\}) \; . \tag{5}$$

□

3.2 Maintaining a Cone ϵ-Pareto Front

The convergence and diversity properties are satisfied by maintaining a Cone ϵ-Pareto front. The convergence property is ensured by storing the nondominated solutions in the archive $\mathcal{H}$. In addition, since each box accommodates only a single vector, the diversity property is also guaranteed.

As we have observed for the ϵ-dominance, the archive update function in the cone ϵ-dominance strategy also uses a two level concept. On the first level, the objective space is discretized into boxes, each box containing a single vector. Applying the cone ϵ-dominance relation at these boxes, the algorithm always maintains a set of non cone ϵ-dominated solutions, ensuring the diversity property. For that, every solution in the archive is assigned a box index ($\boldsymbol{b}$):

$$\boldsymbol{b}_i(\boldsymbol{y}) = \begin{cases} \epsilon_i \left\lfloor y_i/\epsilon_i \right\rfloor \text{ , for minimizing } f_i \\ \epsilon_i \left\lceil y_i/\epsilon_i \right\rceil \text{ , for maximizing } f_i \end{cases} \tag{6}$$

where $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ return, respectively, the closest lower and upper integer to their argument. On the second level, if two vectors share the same box, the former solution is only replaced by a dominating one or by a point closest to the origin of the box, thus guaranteeing convergence. Algorithm 1 summarizes these concepts.

Algorithm 1. Archive update function performed by cone ϵ-dominance.

```
Input: $\mathcal{H}$, $\Psi$, $\boldsymbol{y}$
begin
    if $\boldsymbol{y}$ is cone $\epsilon$-dominated by any $\boldsymbol{y}' \in \mathcal{H}$ then
        Reject $\boldsymbol{y}$;
    else if $\boldsymbol{y}$ shares the same box with an archive member $\boldsymbol{y}'$ then
        if $\boldsymbol{y}$ dominates $\boldsymbol{y}'$ or $\boldsymbol{y}$ is closer to the origin of the box than $\boldsymbol{y}'$ then
            Delete all of the cone $\epsilon$-dominated archive members;
            Replace $\boldsymbol{y}'$ by $\boldsymbol{y}$;
        else
            Reject $\boldsymbol{y}$;
    else if $\boldsymbol{y}$ cone $\epsilon$-dominates any $\boldsymbol{y}' \in \mathcal{H}$ then
        Delete all of the cone $\epsilon$-dominated archive members;
        Insert $\boldsymbol{y}$ into the archive;
    else
        Insert $\boldsymbol{y}$ into the archive;
end
Output: $\mathcal{H}'$
```

3.3 Evaluating the Archive Size

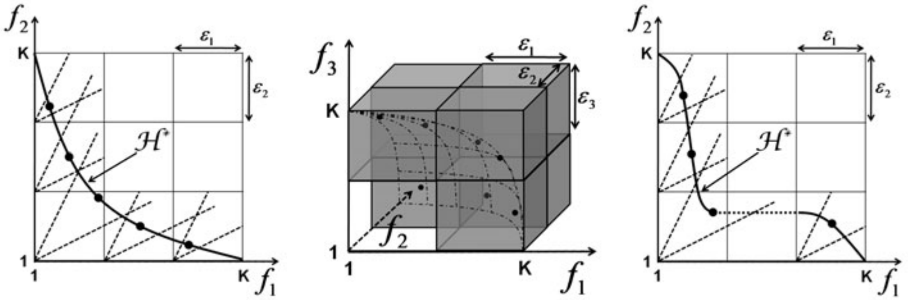
Again, assume that all objectives f_i, $i \in \{1, \ldots, m\}$, are to be minimized, and also that $1 \leq f_i \leq K$, for all i. As observed for the ϵ-dominance, the cone ϵ-dominance approach divides the objective space into $((K-1)/\epsilon)^m$ boxes, and at each box no more than one point can be in $\mathcal{H}$ at the same time. Since the usual dominance relation ensures a monotonic front between the extreme boxes of the hypergrid, the maximum number of boxes that can be "touched" by any front is limited. However, the estimation of feasible solutions inside these touched boxes depends on the connectivity of the Pareto front and also of the κ value.

In general, if any connected monotonic front exists between the extreme boxes of the hypergrid, then the number of boxes that are touched by this front is maximum. Figs. 3(a) and 3(b) illustrate two possible situations in which the number of estimated cone ϵ-Pareto solutions is maximum, i.e., equal to the number of boxes that are touched by the front. For both cases, the number of estimated solutions is five and seven, respectively. These values are calculated as:

$$|\mathcal{H}| \leq m \left[\left(\frac{K-1}{\epsilon} \right)^{m-1} - \left(\frac{K-1}{\epsilon} \right)^{m-2} \right] + 1 \ . \tag{7}$$

Fig. 3(c) presents a possible situation in which a disconnected front has been stated. For this case, the maximum size of $\mathcal{H}$ cannot be reached, however, it is still likely to estimate one solution from each box touched by the front. The ϵ-dominance approach, on the other hand, can only achieve the upper bound for the number of points allowed by a grid when the real Pareto front is linear [5].

Finally, note that the definition in (7) can be generalized by considering a different ϵ for each objective. Observe also that specific bounds on the objective values are not used in the Alg. 1 and are not required to ensure the convergence. They are only employed to demonstrate the relation between ϵ and the size of the archive $\mathcal{H}$.



(a) Connected front.

(b) Concave surface.

(c) Disconnected front.

Fig. 3. Illustration of the relation between ϵ and the size of the archive $\mathcal{H}$

4 Experiments and Validation of the Proposed Approach

In order to validate the proposed cone ϵ-dominance approach, three algorithms are considered for the experimental study: the first employs the standard Pareto dominance relation; the second uses the ϵ-dominance strategy; and the third is implemented by modifying the second one, replacing the ϵ-dominance mechanism by the cone ϵ-dominance approach. This process will enable us to show the performance of the same algorithm with and without cone ϵ-dominance. The three multiobjective evolutionary algorithms are presented next:

1. **NSGA-II:** This algorithm has been proposed by Deb *et al.* [2]. In general terms, the parent and offspring populations are combined and evaluated using the fast nondominated sorting approach, an elitist approach, and an efficient crowding mechanism.
2. **ϵ-MOEA:** This approach has been proposed by Deb *et al.* [7,8], and consists of a steady-state MOEA based on the ϵ-dominance concept introduced in [4]. In this method, two populations (evolutionary population and archive) are evolved simultaneously, and two offspring solutions are created by using one solution from each population. Each offspring is then used to update both parent and archive populations. Note, however, that the archive population is updated based on the ϵ-dominance concept, whereas an usual domination concept is used to update the parent population.
3. **coneϵ-MOEA:** This is a modification of the ϵ-MOEA approach, in which we include cone ϵ-dominance instead of the regular ϵ-dominance concept.

These methods have been implemented, in Matlab, based on the source codes available in [9][3]. Further information about test problems, performance metrics, parameter settings, and statistical design are presented in the following sections.

4.1 Analytical Benchmark Problems

We have chosen five continuous and simple, unconstrained, test problems with different geometrical features for this experimental study. Since the goal here is to investigate the potential advantages of the proposed cone ϵ-dominance approach over the ϵ-dominance strategy, our choice of problems was directed by the geometrical characteristics of the Pareto fronts rather than the difficulty of solving each test problem. In this way, the problems selected are the following: Deb52 [10], Poloni's problem [11], ZDT1 and ZDT6 [12], and DTLZ2 [13]. Table 1 presents further details of these problems. For a future work, we aim at using the proposed strategy to solve more complex multiobjective optimization problems.

Table 1. Analytical test problems adopted in the experimental study

Problem	n	Bounds	Objective functions	Pareto front
P1: Deb52 [10]	2	$[0, 1]$	$f_1(\boldsymbol{x}) = 1 - \exp(-4x_1)\sin^4(10\pi x_1)$ $f_2(\boldsymbol{x}) = g(x_2)h(x_1)$ $g(x_2) = 1 + x_2^2$ $h(x_1) = \begin{cases} 1 - \left(\frac{f_1(\boldsymbol{x})}{g(x_2)}\right)^{10} & \text{if } f_1(\boldsymbol{x}) \leq g(x_2) \\ 0 & \text{otherwise.} \end{cases}$	Concave
P2: Pol [11]	2	$[-\pi, \pi]$	$f_1(\boldsymbol{x}) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2$ $f_2(\boldsymbol{x}) = (x_1 + 3)^2 + (x_2 + 1)^2$ $A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2$ $A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2$ $B_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2$ $B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2$	Nonconvex and disconnected
P3: ZDT1 [12]	30	$[0, 1]$	$f_1(\boldsymbol{x}) = x_1$ $f_2(\boldsymbol{x}) = 1 - \sqrt{x_1/g(\boldsymbol{x})}$ $g(\boldsymbol{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	Convex (multimodal problem)
P4: ZDT6 [12]	10	$[0, 1]$	$f_1(\boldsymbol{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(\boldsymbol{x}) = 1 - (f_1/g(\boldsymbol{x}))^2$ $g(\boldsymbol{x}) = 1 + 9\left[\left(\sum_{i=2}^{n} x_i\right)/(n-1)\right]^{0.25}$	Nonconvex and nonuniformly spaced
P5: DTLZ2 [13]	12	$[0, 1]$	$f_1(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_m))\cos(x_1\pi/2)\cos(x_2\pi/2)$ $f_2(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_m))\cos(x_1\pi/2)\sin(x_2\pi/2)$ $f_3(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_m))\sin(x_1\pi/2)$ $g(\boldsymbol{x}_m) = \sum_{x_i \in \boldsymbol{x}_m} (x_i - 0.5)^2$	Concave surface

4.2 Performance Metrics

Unlike mono-objective optimization, multiobjective optimization techniques are required to consider two different goals: (i) convergence to the Pareto-optimal front, and (ii) maintenance of a diverse set of solutions. To consider this multi-criterion nature in the evaluation of multi-objective algorithms, we have used four different metrics in our analysis.

[3] The implementation of the algorithms, the samples of the true Pareto fronts used, and the raw and processed results of the experiments can be retrieved from the address http://www.cpdee.ufmg.br/~fcampelo/EN/files.html

The first one, the Convergence metric (γ) [2], measures the distance between the obtained nondominated front $\mathcal{H}$ and a detailed sampling of the true Pareto-optimal front $\mathcal{H}^*$:

$$\gamma = \frac{\sum_{i=1}^{|\mathcal{H}|} d_i}{|\mathcal{H}|} \tag{8}$$

where d_i is the Euclidean distance, in the objective space, between the solution $i \in \mathcal{H}$ and the nearest member of $\mathcal{H}^*$. So, the lower the γ value the better the convergence of the solutions in $\mathcal{H}$. Note that a result with $\gamma = 0$ means $\mathcal{H} \subseteq \mathcal{H}^*$, otherwise, $\mathcal{H}$ deviates from $\mathcal{H}^*$.

The second metric, Diversity metric ($\varDelta$) [2], measures the extent of spread achieved among the obtained nondominated solutions in $\mathcal{H}$. By definition:

$$\varDelta = \frac{\sum_{i=1}^{m} d_i^e + \sum_{i=1}^{\mathcal{H}} \left| d_i - \bar{d} \right|}{\sum_{i=1}^{m} d_i^e + |\mathcal{H}|\, \bar{d}} \tag{9}$$

where d_i^e denotes the Euclidean distance between the extreme points in $\mathcal{H}$ and $\mathcal{H}^*$ along the i^{th} coordinate, and d_i measures the Euclidean distance of each point in $\mathcal{H}$ to its closer point in $\mathcal{H}$. So the lower the $\varDelta$ value, the better the distribution of solutions. Notice that a result with $\varDelta = 0$ means the extreme points of $\mathcal{H}^*$ have been found and d_i equals to $\bar{d}$ for all i.

The third metric, S-metric (HV) [3], calculates the hypervolume enclosed by the estimated front $\mathcal{H}$ and a reference point dominated by all solutions in this front. The larger the dominated hypervolume, the better the front is. For all test problems, the reference point has been stated as 10% greater than the upper boundaries of the real Pareto-optimal front. This metric estimates both convergence and diversity of the solutions in $\mathcal{H}$.

Since the power of unary metrics is limited [14], a binary performance metric, the Coverage of Two Sets (CS) [3], has also been considered. This metric quantifies the domination of the final population of one algorithm over another. The CS function is stated as:

$$CS(X', X'') = \frac{|a'' \in X'';\ \exists a' \in X' :\ a' \preceq a''|}{|X''|} \tag{10}$$

where X' and X'' are two sets of objective vectors, and $a' \preceq a''$ means that a' covers a'', that is, either $a' \prec a''$ or $a' = a''$. Function CS maps the ordered pair (X_i, X_j) to the interval $[0, 1]$, in which X_i and X_j denote the final Pareto fronts resulting from algorithm i and j, respectively. The value $CS\,(X_i, X_j) = 1$ implies that all points in X_j are dominated by or equal to points in X_i. The opposite, $CS\,(X_i, X_j) = 0$, represents the situation when none of the points in X_j are covered by the set X_i. Notice that both $CS\,(X_i, X_j)$ and $CS\,(X_j, X_i)$ need to be considered independently since they have distinct meanings.

As is the case of the two first metrics, a detailed sampling of the true Pareto-optimal front of each problem must be known. Since we are dealing with test problems, the true Pareto-optimal front is not difficult to obtain. In this work, we have used uniformly spaced Pareto-optimal solutions as the approximation of the true Pareto-optimal front, which can be retrieved online[3].

4.3 Parameter Settings

To provide a comparison baseline for the performance of the proposed cone ϵ-dominance strategy, some parameter settings were adopted for all the algorithms: population size $N = 100$; probabilities of crossover and mutation $p_{xover} = 1$ and $p_{mut} = 1/n$, respectively; simulated binary crossover (SBX) with parameter $\eta_{xover} = 15$; and polynomial mutation with $\eta_{mut} = 20$. Furthermore, aiming at an archive population size of 100 solutions, we have used the $\boldsymbol{\epsilon}$ values showed in Table 2. The calculated $\boldsymbol{\epsilon}$ values have been obtained following the guidelines provided by Laumanns *et al.* [4] and using (7), for ϵ-MOEA and coneϵ-MOEA, respectively. Since ϵ-dominance and cone ϵ-dominance may lose nondominated points, we have also estimated $\boldsymbol{\epsilon}$ values in order to get roughly 100 solutions in the archive[4]. Both situations are considered in the experimental study. Note also that, for the coneϵ-MOEA, the estimated $\boldsymbol{\epsilon}$ values differ from the calculated ones only for the Pol and DTLZ2 problems, whereas in the ϵ-MOEA all $\boldsymbol{\epsilon}$ values have changed. This phenomenon indicates that the cone ϵ-dominance approach may be less susceptible to the loss of nondominated solutions than the ϵ-dominance.

Table 2. Calculated (calc.) and estimated (est.) $\boldsymbol{\epsilon} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_m\}$ values for a population archive of 100 points

$\boldsymbol{\epsilon}$	Algorithm	Deb52	Pol	ZDT1	ZDT6	DTLZ2
calc.	coneϵ-MOEA	$[0.0164, 0.0198]$	$[0.3168, 0.4950]$	$\epsilon_i = 0.0200$	$[0.0143, 0.0184]$	$\epsilon_i = 0.1595$
	ϵ-MOEA	$[0.0083, 0.0100]$	$[0.1600, 0.2500]$	$\epsilon_i = 0.0100$	$[0.0072, 0.0093]$	$\epsilon_i = 0.1000$
est.	coneϵ-MOEA	$[0.0164, 0.0198]$	$[0.2068, 0.3231]$	$\epsilon_i = 0.0200$	$[0.0143, 0.0184]$	$\epsilon_i = 0.1568$
	ϵ-MOEA	$[0.0031, 0.0030]$	$[0.0390, 0.0380]$	$\epsilon_i = 0.0080$	$[0.0067, 0.0067]$	$[0.06, 0.06, 0.066]$

Since the cone ϵ-dominance approach is influenced by the κ parameter, we perform some preliminary testing to observe the effect of different κ values on the performance of the coneϵ-MOEA. Table 3 shows the effect of this parameter on the values of the unary metrics γ, $\mathit{\Delta}$ and HV for the benchmark problem ZDT1[5]. In this limited test, intermediate values for κ seem to yield reasonably good performance values for all metrics, from which a value of $\kappa = 0.5$ was chosen for all experiments conducted in this work.

4.4 Statistical Design

To evaluate the possible differences between the performance of the methods tested, we have employed tests designed to detect statistically significant differences and to estimate their sizes. The tests were performed independently for each benchmark problem and each of the quality metrics described in the previous Section, and are described in the following paragraphs.

[4] This estimation was performed by testing different $\boldsymbol{\epsilon}$ values in order to get roughly 100 solutions in the archive after 20,000 solution evaluations.

[5] Due to the space restrictions of this paper, only the ZDT1 problem was considered.

Table 3. Influence of different κ values on the performance of the coneϵ-MOEA on test problem ZDT1. Mean and standard deviation over 50 independent runs.

Metric		κ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.9999
γ	Mean	0.0102	0.0039	0.0054	0.0039	0.0036	0.0036	0.0030	0.0029	0.0031	0.0030	0.0030
	SDev	0.0063	0.0029	0.0045	0.0032	0.0043	0.0035	0.0022	0.0031	0.0027	0.0027	0.0026
Δ	Mean	0.2997	0.6022	0.4159	0.1891	0.1659	0.1743	0.1828	0.1905	0.1931	0.1873	0.1911
	SDev	0.0150	0.0369	0.0393	0.0351	0.0215	0.0243	0.0199	0.0201	0.0303	0.0162	0.0260
HV	Mean	0.8447	0.8612	0.8610	0.8655	0.8665	0.8665	0.8675	0.8677	0.8675	0.8675	0.8676
	SDev	0.0101	0.0054	0.0078	0.0057	0.0074	0.0061	0.0038	0.0053	0.0045	0.0047	0.0044
$\lvert\mathcal{H}\rvert$	Mean	36.94	65.58	81.44	97.40	100.52	100.66	100.82	100.78	100.68	100.88	100.64
	SDev	0.3136	3.9646	4.6077	2.6030	0.7351	0.5573	0.4375	0.4647	1.3915	0.3283	2.1262

The raw dataset used for this analysis was composed of the final Pareto-front obtained on 50 independent runs of each algorithm on each problem, from which the quality criteria described earlier were calculated.

The statistical tests were then performed using these calculated values. For the unary metrics (γ, Δ and HV), the null hypotheses used were those of equality of median values, against the two-sided alternative hypotheses. In the particular case of the HV metric, the values obtained by each algorithm were normalized by the hypervolume of the "real" Pareto front available for each problem, in order to analyze the differences between algorithms in terms of percentage gains instead of absolute values.

For the binary metric CS, the hypotheses tested were defined as the difference between the mutual coverages of pairs of algorithms. For instance, the hypotheses for the comparison between the coneϵ-MOEA and the ϵ-MOEA on the Deb52 problem using the CS metric were defined as:

$$\begin{cases} H_0^{P1;CS;(c,\epsilon)} : \tilde{\mu}^{p1}_{CS(c,\epsilon)} - \tilde{\mu}^{p1}_{CS(\epsilon,c)} = 0 \\ H_1^{P1;CS;(c,\epsilon)} : \tilde{\mu}^{p1}_{CS(c,\epsilon)} - \tilde{\mu}^{p1}_{CS(\epsilon,c)} \neq 0 \end{cases} \tag{11}$$

where $\tilde{\mu}^{p1}_{CS(c,\epsilon)}$, $\tilde{\mu}^{p1}_{CS(\epsilon,c)}$ represent the median values of the coverage of the $\underline{\mathbf{c}}$oneϵ-MOEA over the $\underline{\epsilon}$-MOEA (and vice versa) for problem $P1$.

In this work we have employed Wilcoxon's Rank-sum method [15] to test the working hypotheses. Estimations of the effect size were also calculated by means of the Hodges-Lehmann (HL) estimator of the median of differences between two independent samples [16], which represents the median of the set of all possible pairwise differences between two sets of data.

From the definition of the cone ϵ-dominance, it is expected that the performance of the coneϵ-MOEA should be superior to that of the ϵ-MOEA and NSGA-II in those criteria that measure the diversity of solutions over the Pareto front. Metrics where only the convergence to the "true" Pareto front is considered (e.g., the γ metric) should present non-significant or very small differences between the ϵ-MOEA and coneϵ-MOEA, while those measuring only the diversity of points obtained (e.g., the Δ metric) should indicate more significant effects in favor of the coneϵ-MOEA. Hybrid metrics, such as HV and CS, should present mixed results with a tendency to indicate better values for the coneϵ-MOEA over the other two.

5 Results and Discussion

The results obtained for the four performance metrics considered using the calculated and estimated values of ϵ are summarized in Table 4.

First, it is clear that the predicted performance gains for the diversity metric Δ were verified for both the calculated and estimated ϵ. In both situations, the coneϵ-MOEA was able to significantly outperform the NSGA-II in all tested cases, and the ϵ-MOEA in most of the tests (no difference on the DTLZ2 for calculated ϵ, small negative effect on DTLZ2 for estimated ϵ and on ZDT6 in both cases). The size of the positive gains observed for the coneϵ-MOEA were also reasonable, which is in accordance with the predicted behavior.

The expected behavior was also observed for the convergence metric γ. For both the estimated and calculated ϵ, non-significant differences were observed for the algorithm comparisons, and even the statistically significant ones represented very small effects where no tendency could be observed. These results are quite compatible with the idea that the cone ϵ-dominance approach is a diversity-enhancing one, with little or no effect over the ability of the underlying algorithm to converge to the vicinity of the true Pareto-optimal front.

For the hypervolume metric HV, the results were again positive for the coneϵ-MOEA. For the calculated ϵ runs, the cone ϵ-dominance approach outperformed the ϵ-MOEA in all five problems, and the NSGA-II in four. The effect sizes for this metric, which translate as percentual gains, were relatively small, with the largest one being around 8.5%. The same tendency was observed for runs using estimated ϵ values, which indicates that the use of cone ϵ-dominance was able to provide a statistically significant, although small, advantage over the ϵ-dominance in the problems tested, as measured by the HV metric.

The results for the coverage of two sets metric (CS) also indicate an advantage in the use of cone ϵ-dominance. For both the calculated and estimated ϵ, the coneϵ-MOEA was not inferior to the other two algorithms in all cases tested, and presented significant positive effects in most of the cases examined. The effect sizes observed for this metric were also reasonably large, with values up to 0.65.

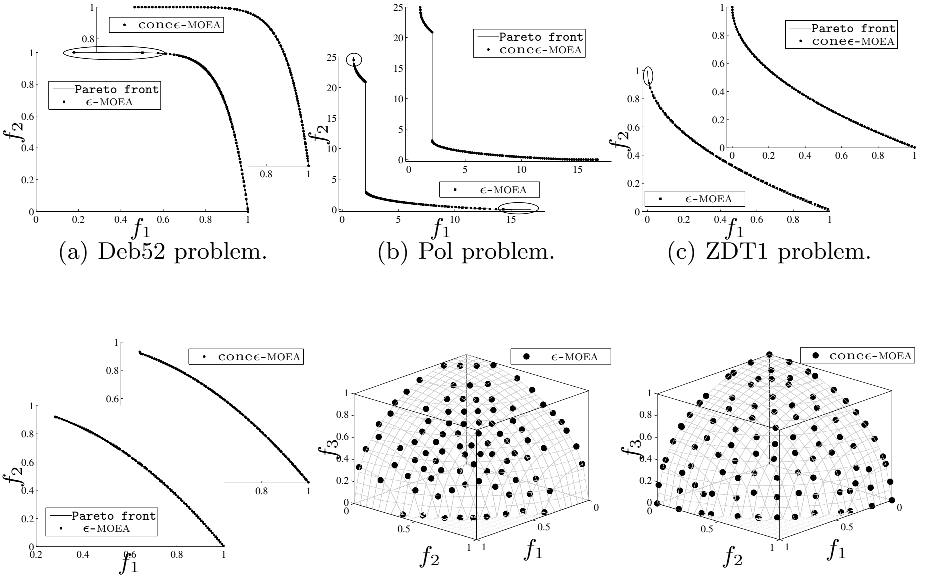
The results obtained for the tests performed indicate that the use of a cone ϵ-dominance criterion can significantly improve the diversity of solutions in the objective space, while leaving other characteristics of the algorithm - such as the computational cost measured by the number of function evaluations, or the ability to converge to the vicinity of the global Pareto front - mostly unchanged. Also, it is important to notice that the calculation of the ϵ value needed to obtain a given size for the Pareto front seems to be much more reliable for the coneϵ-MOEA than the ϵ-MOEA (see Table 2), which also supports the use of the approach based on the cone ϵ-dominance criterion over the pure ϵ-dominance.

Table 4. Comparison between the **C**oneϵ-MOEA, ϵ-MOEA and **N**SGA-II. For each metric, p stands for the p-value obtained using Wilcoxon's Rank-sum test for differences of median values, and HL stands for Hodges-Lehmann estimator of effect size. In the effect size estimation, the algorithms were considered in the order given, and only the (99%) statistically significant effects were calculated. In all comparisons involving the coneϵ-MOEA, *positive* effect sizes indicate a superiority of this approach over the one being compared. Boldface HL values highlight the cases where statistically significant positive effects were observed for the coneϵ-MOEA.

Calculated ϵ											
		Deb52		Pol		ZDT1		ZDT6		DTLZ2	
		p	HL	p	HL	p	HL	p	HL	p	HL
γ	(ϵ, C)	3.4×10^{-11}	-6.5×10^{-5}	9.0×10^{-18}	-2.5×10^{-4}	5.9×10^{-5}	$\mathbf{4.8 \times 10^{-3}}$	6.7×10^{-5}	$\mathbf{1.0 \times 10^{-3}}$	1.8×10^{-17}	$\mathbf{4.9 \times 10^{-3}}$
	(N, C)	5.6×10^{-3}	$\mathbf{1.9 \times 10^{-5}}$	2.2×10^{-1}	-	2.1×10^{-1}	-	2.2×10^{-2}	-	2.8×10^{-8}	$\mathbf{1.7 \times 10^{-3}}$
	(N, ϵ)	9.0×10^{-13}	8.4×10^{-5}	9.0×10^{-18}	2.3×10^{-4}	6.2×10^{-4}	-3.9×10^{-3}	1.6×10^{-9}	-1.3×10^{-3}	5.0×10^{-16}	-3.2×10^{-3}
Δ	(ϵ, C)	7.1×10^{-18}	$\mathbf{5.0 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{5.9 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{1.7 \times 10^{-1}}$	2.6×10^{-5}	-2.6×10^{-2}	3.0×10^{-2}	-
	(N, C)	7.1×10^{-18}	$\mathbf{1.4 \times 10^{-1}}$	1.1×10^{-17}	$\mathbf{2.1 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{2.1 \times 10^{-1}}$	2.9×10^{-17}	$\mathbf{2.0 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{2.7 \times 10^{-1}}$
	(N, ϵ)	7.1×10^{-18}	-3.6×10^{-1}	7.1×10^{-18}	-3.8×10^{-1}	1.6×10^{-7}	4.0×10^{-2}	8.5×10^{-18}	2.3×10^{-1}	7.1×10^{-18}	2.4×10^{-1}
HV	(C, ϵ)	7.1×10^{-18}	$\mathbf{1.1 \times 10^{-2}}$	7.1×10^{-18}	$\mathbf{5.7 \times 10^{-3}}$	7.1×10^{-18}	$\mathbf{1.2 \times 10^{-2}}$	7.1×10^{-18}	$\mathbf{4.1 \times 10^{-3}}$	7.1×10^{-18}	$\mathbf{8.5 \times 10^{-2}}$
	(C, N)	1.3×10^{-16}	$\mathbf{1.7 \times 10^{-3}}$	1.7×10^{-2}	-	1.6×10^{-14}	$\mathbf{2.8 \times 10^{-3}}$	7.1×10^{-18}	$\mathbf{7.0 \times 10^{-4}}$	8.6×10^{-15}	$\mathbf{4.4 \times 10^{-2}}$
	(ϵ, N)	7.1×10^{-18}	-9.2×10^{-3}	7.1×10^{-18}	-6.1×10^{-3}	7.1×10^{-18}	-9.2×10^{-3}	1.3×10^{-2}	-	2.9×10^{-15}	-4.1×10^{-2}
CS	(C, ϵ)	8.3×10^{-8}	0	2.8×10^{-3}	0	1.4×10^{-7}	$\mathbf{6.5 \times 10^{-1}}$	3.3×10^{-6}	$\mathbf{1.4 \times 10^{-1}}$	6.1×10^{-11}	$\mathbf{4.4 \times 10^{-2}}$
	(C, N)	2.9×10^{-20}	$\mathbf{5.0 \times 10^{-2}}$	6.4×10^{-2}	-	1.6×10^{-3}	$\mathbf{1.7 \times 10^{-1}}$	3.1×10^{-1}	-	4.1×10^{-17}	$\mathbf{6.0 \times 10^{-2}}$
	(ϵ, N)	2.7×10^{-17}	4.0×10^{-2}	7.1×10^{-12}	4.0×10^{-2}	3.8×10^{-5}	-4.4×10^{-1}	5.9×10^{-9}	-2.2×10^{-1}	4.1×10^{-8}	-2.4×10^{-2}

Estimated ϵ											
		Deb52		Pol		ZDT1		ZDT6		DTLZ2	
		p	HL	p	HL	p	HL	p	HL	p	HL
γ	(ϵ, C)	1.34×10^{-5}	$\mathbf{1.3 \times 10^{-5}}$	8.0×10^{-18}	-6.3×10^{-5}	2.5×10^{-4}	$\mathbf{3.9 \times 10^{-3}}$	3.7×10^{-5}	$\mathbf{9.0 \times 10^{-4}}$	3.6×10^{-16}	$\mathbf{3.6 \times 10^{-3}}$
	(N, C)	5.64×10^{-3}	$\mathbf{1.9 \times 10^{-5}}$	9.4×10^{-10}	$\mathbf{1.0 \times 10^{-4}}$	2.1×10^{-1}	-	2.2×10^{-2}	-	8.3×10^{-10}	$\mathbf{2.1 \times 10^{-3}}$
	(N, ϵ)	3.8×10^{-1}	-	7.1×10^{-18}	1.6×10^{-4}	1.6×10^{-3}	-3.1×10^{-3}	2.0×10^{-9}	-1.2×10^{-3}	3.2×10^{-9}	-1.5×10^{-3}
Δ	(ϵ, C)	7.1×10^{-18}	$\mathbf{5.1 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{4.6 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{1.7 \times 10^{-1}}$	1.2×10^{-3}	-2.1×10^{-2}	3.3×10^{-6}	-2.6×10^{-2}
	(N, C)	7.1×10^{-18}	$\mathbf{1.4 \times 10^{-1}}$	1.6×10^{-17}	$\mathbf{1.7 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{2.1 \times 10^{-1}}$	2.9×10^{-17}	$\mathbf{2.0 \times 10^{-1}}$	7.1×10^{-18}	$\mathbf{2.2 \times 10^{-1}}$
	(N, ϵ)	7.1×10^{-18}	-3.7×10^{-1}	7.1×10^{-18}	-3.0×10^{-1}	1.1×10^{-7}	4.2×10^{-2}	1.5×10^{-14}	2.3×10^{-1}	7.1×10^{-18}	2.5×10^{-1}
HV	(C, ϵ)	7.1×10^{-18}	-1.5×10^{-3}	7.1×10^{-18}	$\mathbf{1.7 \times 10^{-3}}$	7.1×10^{-18}	$\mathbf{9.0 \times 10^{-3}}$	1.1×10^{-17}	$\mathbf{2.8 \times 10^{-3}}$	3.3×10^{-3}	$\mathbf{1.7 \times 10^{-2}}$
	(C, N)	1.3×10^{-16}	$\mathbf{1.7 \times 10^{-3}}$	2.5×10^{-1}	-	1.6×10^{-14}	$\mathbf{2.8 \times 10^{-3}}$	7.1×10^{-18}	$\mathbf{7.0 \times 10^{-4}}$	4.0×10^{-15}	$\mathbf{4.4 \times 10^{-2}}$
	(ϵ, N)	1.3×10^{-16}	3.3×10^{-3}	7.1×10^{-18}	-1.4×10^{-3}	7.1×10^{-18}	-6.1×10^{-3}	9.3×10^{-1}	-	3.0×10^{-14}	2.6×10^{-2}
CS	(C, ϵ)	1.6×10^{-7}	0	8.0×10^{-7}	-1.0×10^{-2}	2.5×10^{-5}	$\mathbf{4.8 \times 10^{-1}}$	9.3×10^{-4}	$\mathbf{1.8 \times 10^{-1}}$	1.9×10^{-18}	$\mathbf{8.2 \times 10^{-2}}$
	(C, N)	2.9×10^{-20}	$\mathbf{5.0 \times 10^{-2}}$	2.6×10^{-15}	$\mathbf{4.0 \times 10^{-2}}$	1.6×10^{-3}	$\mathbf{1.7 \times 10^{-1}}$	3.1×10^{-1}	-	3.7×10^{-15}	$\mathbf{4.0 \times 10^{-2}}$
	(ϵ, N)	1.6×10^{-18}	6.0×10^{-2}	8.0×10^{-19}	8.5×10^{-2}	1.9×10^{-2}	-	8.0×10^{-8}	-2.1×10^{-1}	6.5×10^{-7}	-1.1×10^{-2}

Figure 4 presents typical Pareto fronts reached for each test problem.



(a) Deb52 problem. (b) Pol problem. (c) ZDT1 problem.

(d) ZDT6 problem. (e) DTLZ2 problem. (f) DTLZ2 problem.

Fig. 4. Efficient solutions generated by coneϵ-MOEA and ϵ-MOEA for the set of estimated $\boldsymbol{\epsilon}$ values. The presented fronts are the outcome of a typical run.

6 Conclusions

We have proposed a relaxed form of Pareto dominance, named cone ϵ-dominance. This approach has been used to ensure both properties of convergence towards the Pareto-optimal front and properties of diversity among the solutions found. Basically, the cone ϵ-dominance strategy takes advantage of the positive aspects of ϵ-dominance, while addressing some of its limitations. As shown, it provides a better control over the resolution of the Pareto found, and also a better spread of solutions along the front.

We have used three evolutionary multiobjective algorithms to evaluate the relative effectiveness of the proposed scheme: NSGA-II, ϵ-MOEA, and coneϵ-MOEA, in which we have included cone ϵ-dominance instead of the regular ϵ-dominance concept. Regarding the performance measures, the experimental results show that the cone ϵ-dominance approach produces statistically competitive results, improving the diversity performance while maintaining the characteristics of convergence toward the Pareto front.

Acknowledgments. This work was supported by the following agencies: National Council for Research and Development (CNPq), grants 306910/2006-3, 141819/2009-0, 305506/2010-2 and 472446/2010-0; Research Foundation of the State of Minas Gerais (FAPEMIG, Brazil), grant Pronex: TEC 01075/09; and

Coordination for the Improvement of Higher Level Personnel (CAPES, Brazil), grant PROCAD 0170/05-4.

References

1. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strengh Pareto Evolutionary Algorithm. Tech. report 103, Computer Engineering and Networks Laboratory (2001)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. Evol. Comp. 6(2), 182–197 (2002)
3. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evol. Comp. 3(4), 257–271 (1999)
4. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization. Evolutionary Computation 10(3), 263–282 (2002)
5. Hernández-Díaz, A.G., Santana-Quintero, L.V., Coello, C.A.C., Molina, J.: Pareto-Adaptive ϵ-Dominance. Evolutionary Computation 15(4), 493–517 (2007)
6. Miettinen, K.M.: Nonlinear Multiobjective Optimization. International Series in Operations Research & Management Science. Springer, Heidelberg (1998)
7. Deb, K., Mohan, M., Mishra, S.: Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In: Fonseca, C.M., et al. (eds.) EMO 2003. LNCS, vol. 2632, pp. 222–236. Springer, Heidelberg (2003)
8. Deb, K., Mohan, M., Mishra, S.: Evaluating the ϵ-Dominance Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. Evolutionary Computation 13(2), 501–525 (2005)
9. Kanpur Genetic Algorithms Laboratory (KanGAL), http://www.iitk.ac.in/kangal/codes.shtml
10. Deb, K.: Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolutionary Computation 7(3), 205–230 (1999)
11. Poloni, C.: Hybrid GA for Multiobjective Aerodynamic Shape Optimization. In: Winter, G., et al. (eds.) Genetic Algorithms in Engineering and Computer Science, pp. 397–416. Wiley, Chichester (1995)
12. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2), 173–195 (2000)
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 105–145. Springer, Heidelberg (2005)
14. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance Assessment of Multiobjective Optimizer: An Analysis and Review. IEEE Trans. Evol. Comp. 7(2), 117–132 (2003)
15. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers, 4th edn. Wiley, Chichester (2006)
16. Hodges, J.L., Lehmann, E.L.: Estimation of location based on ranks. Annals of Mathematical Statistics 34, 598–611 (1963)

Variable Preference Modeling Using Multi-Objective Evolutionary Algorithms

Christian Hirsch, Pradyumn Kumar Shukla, and Hartmut Schmeck

Institute AIFB, Karlsruhe Institute of Technology
Karlsruhe, D-76128, Germany
{c.hirsch,pradyumn.shukla,hartmut.schmeck}@kit.edu

Abstract. Decision making in the presence of multiple and conflicting objectives requires preference from the decision maker. The decision maker's preferences give rise to a domination structure. Till now, most of the research has been focussed on the standard domination structure based on the Pareto-domination principle. However, various real world applications like medical image registration, financial applications, multi-criteria n-person games, among others, or even the preference model of decision makers frequently give rise to a so-called *variable* domination structure, in which the domination itself changes from point to point. Although variable domination is studied in the classical community since the early seventies, we could not find a single study in the evolutionary domain, even though, as the results of this paper show, multi-objective evolutionary algorithms can deal with the vagaries of a variable domination structure. The contributions of this paper are multiple-folds. Firstly, the algorithms are shown to be able to find a well-diverse set of the optimal solutions satisfying a variable domination structure. This is shown by simulation results on a number of test-problems. Secondly, it answers a hitherto open question in the classical community to develop a numerical method for finding a well-diverse set of such solutions. Thirdly, theoretical results are derived which facilitate the use of an evolutionary multi-objective algorithm. The theoretical results are of importance on their own. The results of this paper adequately show the niche of multi-objective evolutionary algorithms in variable preference modeling.

1 Introduction

A variety of complex decision making problems in engineering and mathematical applications are usually multi-objective in nature. The objective can be time, cost, safety, performance among others. A multi-objective optimization problem is characterized by multiple and conflicting objective functions $F_1, \ldots, F_m : \mathbb{R}^n \to \mathbb{R}$. Decision making with these kind of problems requires the elicitation of preferences from the decision maker. The decision maker's preferences give rise to a domination structure. For example, the decision maker can provide a preference model which characterizes the set of bad/ dominated directions in the space of the objectives. This preference model could consist of domination and/ or preference cones. Till now, most of the research has been on the standard

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 91–105, 2011.

domination structure based on the Pareto-domination principle or using a polyhedral cone which can include some desirable trade-off among the objectives [1].

Yu in his seminal work [2] was the first to propose to use variable cones in preference modeling. In a practical context, this says that preference depend upon the current point in the objective space (so called *decisional wealth* in [3]). Variable domination cones are found to be useful in various applications. For example, they have found a recent place in medical engineering where the aim is to merge different medical images obtained by different methods (say computer tomography, ultrasound, positron emission tomography among others). In the problem of medical image registration, one searches for a best transformation map. The variable domination cone used in [4, 5] depends upon the point $\mathbf{w} \in \mathbb{R}^m$ and is defined by

$$\mathcal{C}(\mathbf{w}) := \{\mathbf{d} \in \mathbb{R}^m : \sum_{i=1}^{m} \operatorname{sgn}(d_i)\mathbf{w}_i \geq 0\},$$

where

$$\operatorname{sgn}(d_i) := \begin{cases} 1 & \text{if } d_i > 0; \\ -1 & \text{if } d_i < 0; \\ 0 & \text{if } d_i = 0. \end{cases}$$

This cone is in general non-convex and obviously an example of variable domination structure. These concepts are applied, for example, in multi-objective n-person cooperative as well as noncooperative games [6], in general resource allocation models and location theory [7, 8]. We do not go in further application details here and refer the reader to the original studies.

Problems with variable domination structure give rise to two different types of optimal solutions: minimal points and nondominated points. Both these types are defined in the next section. Here, we only mention that these points lie in the objective space and they are not to be confused with the standard terminology of Pareto-optimal/ nondominated/ efficient points [9].

In classical literature [4, 10], we do find some scalarization techniques to find *one* minimal/ nondominated point. However, there have been few attempts to develop algorithms for finding a representative subset of the minimal/ nondominated set. This problem has also been highlighted in a recent study [4]. Finding multiple solutions in a single run has been a niche of multi-objective evolutionary algorithms [1, 11]. In this paper, we develop algorithms for finding a well-diverse subset of the minimal/ nondominated set. To the best of our knowledge, this is the first study which proposes a multi-objective evolutionary (NSGA-II based) approach for this. In addition, we discuss a theoretical justification for our evolutionary approach.

This paper is divided into four sections of which this is the first. The next section presents the variable domination structure and some theoretical results. The third section presents extensive simulation results using NSGA-II based algorithms. Conclusions as well as extensions which emanated from this study are presented at the end of this contribution.

2 Preliminaries and Theoretical Results

A multi-objective optimization problem (MOP, in short) can be mathematically stated as follows:

$$\begin{aligned} \text{minimize} \;\; & F(\mathbf{x}) := (F_1(\mathbf{x}), F_2(\mathbf{x}), \ldots, F_m(\mathbf{x})) \\ \text{subject to} \;\; & \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^n \end{aligned}$$

where $\mathbf{X}$ is the set of feasible alternatives and $F_1(\mathbf{x}), F_2(\mathbf{x}), \ldots, F_m(\mathbf{x})$ are m (at least two) objective functions that need to be minimized. The spaces $\mathbf{X}$ and $F(\mathbf{X}) := \{F(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$ are also called the *decision space* and the *objective space*, respectively.

Let the non-negative orthant of $\mathbb{R}^m$ be denoted by $\mathbb{R}^m_+$, i.e.,

$$\mathbb{R}^m_+ := \{\mathbf{y} \in \mathbb{R}^m : y_i \geq 0, \text{for all } i = 1, 2, \ldots, m\}.$$

A nonempty set $\mathcal{C} \subseteq \mathbb{R}^m$ is called a *cone* if $c \in \mathcal{C} \Rightarrow \lambda c \in \mathcal{C}$ for all $\lambda \geq 0$. A cone $\mathcal{C}$ is convex if $\mathcal{C}+\mathcal{C} \subset \mathcal{C}$. A cone $\mathcal{C}$ is called pointed if it satisfies that $\mathcal{C} \cap (-\mathcal{C}) = \{\mathbf{0}\}$, where $\mathbf{0}$ is the zero vector in $\mathbb{R}^m$. Hence, $\mathbb{R}^m_+$ is a closed, convex and pointed cone and in multi-objective community it is commonly known as the *Pareto-cone.* Using the Pareto-cone, we can define a *Pareto-domination structure* on the objective space. This structure allows us to compare vectors in the objective space as follows. We say that a vector $\mathbf{u} := (u_1, u_2, \ldots, u_m)$ *Pareto-dominates* a vector $\mathbf{v} := (v_1, v_2, \ldots, v_m)$ if $u_i \leq v_i$ for all $i = 1, 2, \ldots, m$ and $\mathbf{u} \neq \mathbf{v}$. If neither $\mathbf{u}$ Pareto-dominates $\mathbf{v}$ nor $\mathbf{v}$ Pareto-dominates $\mathbf{u}$, we call $\mathbf{u}$ and $\mathbf{v}$ *Pareto-nondominated* to each other. Moreover, we call the Pareto-domination structure as the *standard domination structure* as it is the most-widely used in multi-objective optimization problems (see [1, 9]).

Definition 1 (Pareto-optimal point). *A point* $\hat{\mathbf{x}} \in \mathbf{X}$ *is called* Pareto-optimal *if no other point in* $\mathbf{X}$ *Pareto-dominates it. Equivalently, a point* $\hat{\mathbf{x}} \in \mathbf{X}$ *is* Pareto-optimal *if and only if*

$$\left(\{F(\hat{\mathbf{x}})\} - \mathbb{R}^m_+\right) \cap F(\mathbf{X}) = \{F(\hat{\mathbf{x}})\}.$$

Let $\mathbf{X}_p$ and $\mathcal{E} := F(\mathbf{X}_p)$ denote the set of Pareto-optimal points and the set of Pareto-efficient points, respectively. The set $\mathcal{E}$ is the image of the set of Pareto-optimal points in the objective space (i.e., $\mathcal{E} \subset \mathbb{R}^m$).

Yu [12] proposed to use a constant convex cone to model decision makers' preferences. Using a cone $\mathcal{C} \subseteq \mathbb{R}^m$, he defined a different domination structure as follows: the vector $\mathbf{u}$ *$\mathcal{C}$-dominates* the vector $\mathbf{v}$, if $\mathbf{v} - \mathbf{u} \in \mathcal{C} \setminus \{\mathbf{0}\}$. Unless stated otherwise, we assume throughout this paper that all the cones are closed, convex, and pointed. Moreover, we also assume that the Pareto-cone is always contained in (i.e., is a subset of) all the cones and that the set $F(\mathbf{X})$ is compact.

Definition 2 ($\mathcal{C}$-optimal point). *A point* $\hat{\mathbf{x}} \in \mathbf{X}$ *is called* $\mathcal{C}$-optimal *if no other point in* $\mathbf{X}$ *$\mathcal{C}$-dominates it. Equivalently, a point* $\hat{\mathbf{x}} \in \mathbf{X}$ *is* $\mathcal{C}$-optimal *if and only if*

$$(\{F(\hat{\mathbf{x}})\} - \mathcal{C}) \cap F(\mathbf{X}) = \{F(\hat{\mathbf{x}})\}.$$

Obviously, we get the standard domination structure if $\mathcal{C} := \mathbb{R}^m_+$. The cone $\mathcal{C}$ can be non-convex as well. MOPs with a cone which might be non-convex are discussed in [13, 14, 15].

In [3], it is noticed that the importance of objective functions may change during the decision making process depending upon the current objective function value. Some examples of this are also given in [16]. A constant cone is not useful to compare objective function values in such cases. Due to this limitation of a constant cone, several attempts have been made in the classical literature to extend dominance ideas using *variable cones*. Basically, a variable cone is a cone that is a function of a point in the objective space. Hence, there is a cone $\mathcal{C}(\mathbf{y})$ associated with each point $\mathbf{y}$ in the objective space. Using a variable cone, we can define two different domination relations ($\leq_1$ and $\leq_2$) by

$$\mathbf{u} \leq_1 \mathbf{v} \text{ if } \mathbf{v} \in \{\mathbf{u}\} + \mathcal{C}(\mathbf{v}) \tag{1}$$

and

$$\mathbf{u} \leq_2 \mathbf{v} \text{ if } \mathbf{v} \in \{\mathbf{u}\} + \mathcal{C}(\mathbf{u}), \tag{2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are vectors in the objective space. The domination structures given by (1) and (2) are termed *variable domination structures*. Note that (1) and (2) are the same if the cones are not variable (in this case it is equivalent to $\mathcal{C}$-domination).

The domination relation (1) and (2) lead to two optimality notions.

Definition 3. *A point* $\hat{\mathbf{u}} \in F(\mathbf{X})$ *is called a* minimal point *of* $F(\mathbf{X})$ *if*

$$(\{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}})) \cap F(\mathbf{X}) = \{\hat{\mathbf{u}}\}. \tag{3}$$

Equivalently, if there is no $\mathbf{v} \in F(\mathbf{X})$ *such that*

$$\mathbf{v} \in \{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}}) \setminus \{\mathbf{0}\}, \tag{4}$$

then $\hat{\mathbf{u}} \in F(\mathbf{X})$ *is a* minimal point *of* $F(\mathbf{X})$. *The set of all minimal elements is called the* minimal-set *and is denoted by* $\mathcal{E}_\mathcal{M}$.

Definition 4. *A point* $\hat{\mathbf{u}} \in F(\mathbf{X})$ *is called a* nondominated point *of* $F(\mathbf{X})$ *if there is* no $\mathbf{v} \in F(\mathbf{X})$ *such that*

$$\hat{\mathbf{u}} \in \{\mathbf{v}\} + \mathcal{C}(\mathbf{v}) \setminus \{\mathbf{0}\}. \tag{5}$$

The set of all nondominated elements is called the nondominated-set *and is denoted by* $\mathcal{E}_\mathcal{N}$.

Let $\mathbf{X}_\mathcal{M}$ and $\mathbf{X}_\mathcal{N}$ denote the pre-images (the points in the decision space) of the minimal and the nondominated sets respectively. This means that $\mathcal{E}_\mathcal{M} = F(\mathbf{X}_\mathcal{M})$ and $\mathcal{E}_\mathcal{N} = F(\mathbf{X}_\mathcal{N})$. The concept of minimal elements is also described in [17, 18, 19, 10, 20, 4]. The concept of nondominated elements is based on [20, 12, 15, 21, 4]. The nondominated-set and the minimal-set are in general different. It is easy to verify that both of these sets are equal if a constant cone is used, i.e. if $\mathcal{C}(\mathbf{y}) := \mathcal{C}$. It is not only that Definitions 3 and 4 (and the above concepts of variable domination structure) exist in a mathematical realm, these concepts have (mathematical and) real-world applications (see Section 1).

The next lemma shows that under the assumptions of this paper, both nondominated and minimal points belong to the usual Pareto-efficient front.

Lemma 1. $\mathbf{X}_{\mathcal{M}} \subseteq X_p$*,* $\mathbf{X}_{\mathcal{N}} \subseteq X_p$*,* $\mathcal{E}_{\mathcal{M}} \subseteq \mathcal{E}$ *and* $\mathcal{E}_{\mathcal{N}} \subseteq \mathcal{E}$*.*

Proof: As we assumed throughout that all the cones contain the Pareto-cone, the result follows easily from [4, Lemma 2.13 (a)]. □

Lemmas 2 and 3 are motivated by practical considerations, to reduce the computation effort of a population based multi-objective algorithm for finding minimal/ nondominated points.

Lemma 2. $\hat{\mathbf{u}} \in F(\mathbf{X})$ *is a minimal point of* $F(\mathbf{X})$ if and only if $\hat{\mathbf{u}} \in \mathcal{E}$ *and* $\hat{\mathbf{u}}$ *is a minimal point of* $\mathcal{E}$*.*

Proof: ($\Rightarrow$) Let $\hat{\mathbf{u}} \in F(\mathbf{X})$ be a minimal point of $F(\mathbf{X})$. Hence, $(\hat{\mathbf{u}} - \mathcal{C}(\hat{\mathbf{u}})) \cap F(\mathbf{X}) = \{\hat{\mathbf{u}}\}$. $\mathcal{E} \subseteq F(\mathbf{X})$ together with Lemma 1 gives that $\hat{\mathbf{u}} \in \mathcal{E}$ and moreover,

$$(\{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}})) \cap \mathcal{E} = \{\hat{\mathbf{u}}\}.$$

This obviously means that there is no $\mathbf{v} \in \mathcal{E}$ such that $\mathbf{v} \in \{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}}) \setminus \{\mathbf{0}\}$ and hence, from Definition 3 (replacing $F(\mathbf{X})$ by $\mathcal{E}$) we obtain that $\hat{\mathbf{u}} \in \mathcal{E}$ is a minimal point of $\mathcal{E}$.

($\Leftarrow$) Let $\hat{\mathbf{u}} \in F(\mathbf{X})$ be a minimal point of $\mathcal{E}$. Thus, there is no $\mathbf{v} \in \mathcal{E}$ such that $\mathbf{v} \in \{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}}) \setminus \{\mathbf{0}\}$.

In order to show that $\hat{\mathbf{u}}$ is also a minimal point of $F(\mathbf{X})$, we take an arbitrary but fixed element $\mathbf{w} \in F(\mathbf{X}) \setminus \mathcal{E}$ and assume that

$$\mathbf{w} \in \{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}}) \setminus \{\mathbf{0}\}. \tag{6}$$

Now, as $\mathbf{w} \notin \mathcal{E}$ and as $F(\mathbf{X})$ is assumed to be compact (throughout this paper), we get the existence of a $\hat{\mathbf{w}} \in \mathcal{E}$ which Pareto-dominates it, i.e., $\hat{\mathbf{w}} \in \{\mathbf{w}\} - \mathbb{R}^m_+ \setminus \{\mathbf{0}\}$. This together with (6) and the convexity of $\mathcal{C}(\hat{\mathbf{u}})$, gives that

$$\hat{\mathbf{w}} \in \{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}}) \setminus \{\mathbf{0}\} - \mathbb{R}^m_+ \setminus \{\mathbf{0}\} \subseteq \{\hat{\mathbf{u}}\} - \mathcal{C}(\hat{\mathbf{u}}) \setminus \{\mathbf{0}\}.$$

Using the last inclusion and that $\hat{\mathbf{w}} \in \mathcal{E}$, we arrive at a contradiction to the minimality of $\hat{\mathbf{u}}$ in $\mathcal{E}$. Hence the statement of the lemma follows. □

Assumption 1. *The domination structure is such that if* $\mathbf{u}$ *Pareto-dominates* $\mathbf{v}$*, then* $\mathcal{C}(\mathbf{v}) \subseteq \mathcal{C}(\mathbf{u})$*.*

Lemma 3. *Let Assumption 1 hold. Then,* $\hat{\mathbf{u}} \in F(\mathbf{X})$ *is a nondominated point of* $F(\mathbf{X})$ if and only if $\hat{\mathbf{u}} \in \mathcal{E}$ *and* $\hat{\mathbf{u}}$ *is a nondominated point of* $\mathcal{E}$*.*

Proof: ($\Rightarrow$) Let $\hat{\mathbf{u}} \in F(\mathbf{X})$ be a nondominated point of $F(\mathbf{X})$. Definition 4 gives that there is *no* $\mathbf{v} \in F(\mathbf{X})$ such that

$$\hat{\mathbf{u}} \in \{\mathbf{v}\} + \mathcal{C}(\mathbf{v}) \setminus \{\mathbf{0}\}. \tag{7}$$

This obviously means that there is no $\mathbf{v} \in \mathcal{E}$ such that $\hat{\mathbf{u}} \in \{\mathbf{v}\} + \mathcal{C}(\mathbf{v}) \setminus \{\mathbf{0}\}$ and hence, from Definition 4 (replacing $F(\mathbf{X})$ by $\mathcal{E}$) we obtain that $\hat{\mathbf{u}} \in \mathcal{E}$ is a nondominated point of $\mathcal{E}$.

($\Leftarrow$) Let $\hat{\mathbf{u}} \in F(\mathbf{X})$ be a nondominated point of $\mathcal{E}$. Thus, there is no $\mathbf{v} \in \mathcal{E}$ such that $\hat{\mathbf{u}} \in \{\mathbf{v}\} + \mathcal{C}(\mathbf{v}) \setminus \{\mathbf{0}\}$. This happens if and only if

$$\hat{\mathbf{u}} \notin \{\mathbf{v}\} + \mathcal{C}(\mathbf{v}) \setminus \{\mathbf{0}\} \quad \text{for all } \mathbf{v} \in \mathcal{E}. \tag{8}$$

We will show that the set $\mathcal{E}$ in (8) can be replaced by $F(\mathbf{X})$. Taking an arbitrary but fixed element $\mathbf{w} \in F(\mathbf{X}) \setminus \mathcal{E}$ and assuming that $\mathbf{w} \leq_2 \hat{\mathbf{u}}$ we get the following:

$$\hat{\mathbf{u}} \in \{\mathbf{w}\} + \mathcal{C}(\mathbf{w}) \setminus \{\mathbf{0}\}. \tag{9}$$

Now, as $\mathbf{w} \notin \mathcal{E}$ and as $F(\mathbf{X})$ is assumed to be compact we get the existence of a $\hat{\mathbf{w}} \in \mathcal{E}$ which Pareto-dominates $\mathbf{w}$, i.e., $\mathbf{w} \in \{\hat{\mathbf{w}}\} + \mathbb{R}^m_+ \setminus \{\mathbf{0}\}$. This together with (9), the convexity of $\mathcal{C}(\mathbf{w})$, and Assumption 1 gives

$$\begin{aligned}\hat{\mathbf{u}} \in \{\mathbf{w}\} + \mathcal{C}(\mathbf{w}) \setminus \{\mathbf{0}\} &\subseteq \{\hat{\mathbf{w}}\} + \mathbb{R}^m_+ \setminus \{\mathbf{0}\} + \mathcal{C}(\mathbf{w}) \setminus \{\mathbf{0}\} \\ &\subseteq \{\hat{\mathbf{w}}\} + \mathcal{C}(\mathbf{w}) \setminus \{\mathbf{0}\} \\ &\subseteq \{\hat{\mathbf{w}}\} + \mathcal{C}(\hat{\mathbf{w}}) \setminus \{\mathbf{0}\}.\end{aligned}$$

Therefore, we obtain that $\hat{\mathbf{u}} \in \{\hat{\mathbf{w}}\} + \mathcal{C}(\hat{\mathbf{w}}) \setminus \{\mathbf{0}\}$ for $\hat{\mathbf{w}} \in \mathcal{E}$, and we arrive at a contradiction. Hence the statement of the lemma follows. □

Remark 1. Note that while Lemma 2 (corresponding to minimal points), does not require Assumption 1, this assumption is crucial for Lemma 3 (corresponding to nondominated points) to hold. However, Assumption 1 seems reasonable. A larger cone dominates a larger region and if $\mathbf{u}$ Pareto-dominates $\mathbf{v}$, then the dominated region by the point $\mathbf{u}$ should intuitively be larger ($\mathcal{C}(\mathbf{v}) \subseteq \mathcal{C}(\mathbf{u})$). This assumption is in line with the Pareto dominance compliant mechanism (see the discussion in [22]).

Remark 2. It is important to highlight the importance of Lemmas 2 and 3. They show that in order to check if a point is minimal or nondominated, it is sufficient to check the variable cone conditions w.r.t. Pareto-optimal points only. For any algorithm, this would drastically reduce the computational effort.

Usually, ranking based on finding the fronts in NSGA-II (or finding the set $\mathcal{E}$) is the most time consuming step. If one uses the Pareto-domination structure, there are some efficient divide and conquer based approaches for reducing the complexity of rankings (see [23, 24]). These fast approaches employ sorting of the first objective. However, for non-standard domination structures like variable domination, or $\mathcal{C}$-domination, there is no fast algorithm available and we think that the complexity of doing a variable domination sorting might not be possible to reduce (sorting of the objectives gives no relevant information here). Hence, the additional computational burden of finding minimal/ nondominated points should be reduced as far as possible. Here, the theoretical results of Lemmas 2 and 3 come to our rescue. Finding a fast algorithm for sorting based on $\mathcal{C}$ (or on variable) domination structures is a relevant open question.

3 Experimental Study

In this section, we propose two NSGA-II based algorithms for solving multi-objective problems having a variable domination structure. Moreover, we present simulation results on a number of test-problems.

3.1 Algorithms and Experimental Setup

The first algorithm for such problems is called vNSGA-II. It is the usual NSGA-II using a variable domination instead of the usual Pareto-domination. For finding minimal elements and nondominated points, vNSGA-II uses the two different domination relations ($\leq_1$ and $\leq_2$) defined by (1) and (2), respectively. These domination relations are used throughout in vNSGA-II instead of the usual Pareto-domination. The naive method for ranking (comparing every point with every other using variable domination) is used here (see Remark 2). In the past, NSGA-II has been tested with Pareto and polyhedral-cone domination structures (the so called guided domination approach [25]) but this is the first time that a variable and nonlinear cone domination is used to guide the search.

The second algorithm (for problems with variable domination) is called vPNSGA-II. It uses the general framework that was introduced in [26] and works as follows. In standard NSGA-II, the parent population and the offspring population are combined and a sorting based on Pareto-domination is done. This sorting gives fronts $\mathcal{F}_1, \ldots, \mathcal{F}_r$. We do the same in vPNSGA-II, however from $\mathcal{F}_1$ (best front), we additionally create a set $\mathcal{F}_0$ whose members satisfy the appropriate domination relations ($\leq_1$ or $\leq_2$ whatever is used). The domination relations are checked among the front $\mathcal{F}_1$ members only. This is theoretically justified by Lemmas 2 and 3 which state that the question whether a point is minimal (nondominated) or not can be answered by checking the variable domination requirement from the nondominated front only. Whenever the set $\mathcal{F}_0$ is non-empty, we have this additional front. Members of $\mathcal{F}_0$ are assigned a rank of 0 (better than 1, the rank of $\mathcal{F}_1$, as we consider minimization in the tournament selection). This modified ranking guides the search in an appropriate direction.

The variable domination cone that we use in this study is taken from Engau[10]. He proposed a family of so-called *Bishop-Phelps cones* (that have many applications in nonlinear analysis and multi-objective applications). These cones are described by two parameters: a scalar γ which controls the angle of the cone (in $\mathbb{R}^m$) and a vector $\mathbf{p} \in \mathbb{R}^m$. Based on these parameters, the variable domination cone $\mathcal{C}(\mathbf{u})$ is defined by

$$\mathcal{C}(\mathbf{u}) := \{\mathbf{d} | \langle \mathbf{d}, \mathbf{u} - \mathbf{p} \rangle \geq \gamma \cdot \|\mathbf{d}\| \cdot [\mathbf{u} - \mathbf{p}]_{\min}\}\,, \tag{10}$$

where $[\mathbf{u} - \mathbf{p}]_{\min}$ denotes the minimal component of the vector $\mathbf{u} - \mathbf{p}$. The reference point p is usually taken as the ideal vector, i.e., the vector having the i^{th} component as $\inf\{\mathbf{v}_i | \mathbf{v} \in F(\mathbf{X})\}$. The parameter $\gamma \in (0, 1]$ ensures that the Pareto cone $\mathbb{R}^m_+ \subseteq \mathcal{C}(\mathbf{u})$. Although $\mathcal{C}(\mathbf{u})$ is defined for arbitrary $\gamma \in \mathbb{R}$ and any point $\mathbf{p} \in \mathbb{R}^m$ instead of the ideal point, the above conditions guarantee that $\mathcal{C}(\mathbf{u})$ is closed, convex and pointed.

We test vNSGA-II and vPNSGA-II on 34 test problem instances. The optimal fronts corresponding to the variable domination cone $\mathcal{C}(\mathbf{u})$ is shown in Figures 1–14. As can be seen from the figures, the optimal fronts are a subset of the efficient front. The test problems chosen are of varying complexity and are from different test suites that we find in literature. These include two problems from the *CTP suite* (CTP1, CTP7) [27], two from the *DTLZ suite* (DTLZ7, DTLZ8, 3 objectives) [28], two from the *CEC-2007 competition* (SZDT1, SZDT2), four from the *WFG suite* (WFG1, WFG2, with both 2 and 3 objectives) [29] and six from the *ZDT suite* (ZDT1, ZDT2, ZDT3, ZDT4, ZDT5, ZDT6) [1]. We note that this paper is among the very few studies that consider CTP problems and additionally ZDT5, a difficult discrete problem. For all the problems, we use the zero vector as the ideal point, and use the two values 0.5 and 1.0 for γ. Additionally, we took the ideal point $(-1, -1)$ for ZDT3. Including five existing test suites, this study is among the most comprehensive experimentations that we find in literature.

Note that the use of variable domination cones restricts the set $\mathcal{E}$ of all these problems (see Lemma 1). For all problems, we compute a well-distributed approximation of the preferred (minimal and nondominated) set as follows. Corresponding to a problem, we first generate 5,000 well-diverse points on the efficient front. From these points we calculate the minimal and nondominated points, i.e., the points that satisfy the $\leq_1$ and $\leq_2$ domination relations (note that from Lemma 2, we only need to consider X_p for this).

In order to evaluate the results, we use the Inverted generational distance (IGD) and Generational distance (GD) metrics (w.r.t. the obtained reference set). For statistical evaluation, we run each algorithm for 51 times and present various summary statistics. Moreover, we also use the attainment surface based statistical metric described in [30] and the median (50%) attainment surface (26^{st}) is plotted.

For all problems solved, we use a population of size 100 and set the maximum number of function evaluations as 20,000 (200 generations). We use a standard real-parameter SBX and polynomial mutation operator with $\eta_c = 15$ and $\eta_m = 20$, respectively [1]. For ZDT5, we use a bit-flip mutation operator and a single point crossover. The source code of both vNSGA-II and vPNSGA-II (in C++) is made available[1]. The data files for all the 51 runs of all the problems are available on request. This would benefit any other classical or evolutionary study on multi-objective problems with a variable non-domination structure.

[1] http://www.aifb.kit.edu/web/vNSGA-II/en

3.2 Simulation Results

In Tables 1, 2, 3 and 4, the entries of the algorithms yielding the best results are highlighted in a light gray. Tables 1 and 2 present, for all problems, the median and the interquartile range (in subscript) for the GD metric, using $\gamma = 0.5$ and $\gamma = 1.0$, respectively. The first row entry denotes the algorithm working on to find $\mathcal{E}_{\mathcal{M}}$ and $\mathcal{E}_{\mathcal{N}}$ sets. For example, vNSGA-II ($\mathcal{E}_{\mathcal{N}}$) denotes the vNSGA-II working with the $\leq_2$ domination relation to find $\mathcal{E}_{\mathcal{N}}$. The problem ZDT3-1 is ZDZ3 with $\mathbf{p}=(-1,-1)$. For the other problems we use $\mathbf{p}=(0,0)$.

From Tables 1 and 2 we see that vNSGA-II outperforms vPNSGA-II, when $\gamma = 0.5$. A small value of γ in (10) leads to a larger dominated area which in turn implies that the corresponding solution set is reduced. The variable domination structure is used throughout in vNSGA-II and this gives a more accurate guidance towards the smaller set. On the other hand, vPNSGA-II changes the ranking by creating a preferred front out of the best front (in the sense of Pareto-domination), and hence might not work on problems where the Pareto-optimal front is much bigger than the set of minimal or nondominated elements (which happens if a small value of γ is chosen). Both the algorithms work equally well for a large γ. Tables 3 and 4, present the corresponding results for the IGD metric. The interpretation of the results is similar to Tables 1 and 2. The boxplots in Figures 15 and 16 show a great variability in the observations. It is worthwhile to notice that finding $\mathcal{E}_{\mathcal{N}}$ and $\mathcal{E}_{\mathcal{M}}$ for ZDT5 is the most difficult (worst GD and IGD values).

Figure 1 shows the plot of $\mathcal{E}_{\mathcal{N}}$ and sample runs of vNSGA-II and vPNSGA-II on the convex problem SZDT1. We see that both the algorithms have no difficulty in finding the entire set $\mathcal{E}_{\mathcal{N}}$. However, the attainment surface plot in Figure 2 shows that vPNSGA-II performs slightly better. Next, we consider the non-convex test problems ZDT2 and the shifted version SZDT2. Here, the variable

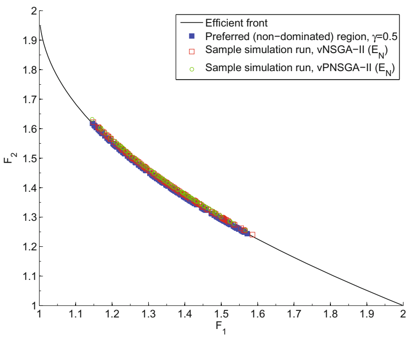


Fig. 1. $\mathcal{E}_{\mathcal{N}}$ and sample runs of vNSGA-II and vPNSGA-II on SZDT1

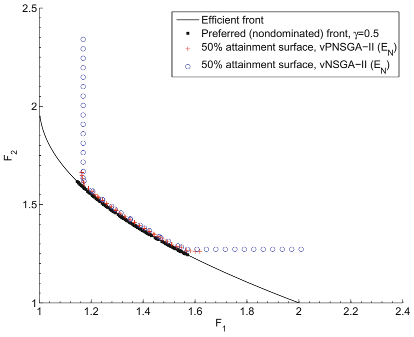


Fig. 2. $\mathcal{E}_{\mathcal{N}}$ and median attainment surface plot for SZDT1

Table 1. GD. Median and interquartile range, for problems with $\gamma = 0.5$

	vNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vNSGA-II ($\mathcal{E}_{\mathcal{N}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{N}}$)
ZDT1	$0.000041_{0.000003}$	$0.000047_{0.000003}$	$0.000043_{0.000005}$	$0.000049_{0.000020}$
ZDT2	$0.007437_{0.000383}$	$0.007112_{0.000655}$	$0.001033_{0.001423}$	$0.001512_{0.001652}$
ZDT3	$0.000056_{0.000024}$	$0.000424_{0.000178}$	$0.000277_{0.000011}$	$0.000336_{0.000081}$
ZDT3-1	$0.000066_{0.000009}$	$0.000211_{0.000091}$	$0.000100_{0.000005}$	$0.000144_{0.000036}$
ZDT4	$0.000328_{0.000404}$	$0.000491_{0.000426}$	$0.000451_{0.000407}$	$0.000527_{0.000644}$
ZDT5	$0.416340_{0.016336}$	$0.457126_{0.017216}$	$0.412098_{0.012548}$	$0.449668_{0.019501}$
ZDT6	$0.002037_{0.000190}$	$0.000994_{0.000212}$	$0.001037_{0.002056}$	$0.000775_{0.000559}$
SZDT1	$0.000591_{0.000117}$	$0.000788_{0.000170}$	$0.000615_{0.000145}$	$0.000803_{0.000185}$
SZDT2	$0.005970_{0.000390}$	$0.005707_{0.002231}$	$0.001059_{0.000238}$	$0.001432_{0.000322}$
CTP1	$0.000071_{0.000021}$	$0.000071_{0.000017}$	$0.000062_{0.000084}$	$0.000085_{0.000112}$
CTP7	$0.000010_{0.000001}$	$0.000010_{0.000001}$	$0.000007_{0.000001}$	$0.000007_{0.000000}$
WFG1-2D	$0.089741_{0.005875}$	$0.345861_{0.175960}$	$0.089030_{0.007313}$	$0.165391_{0.033064}$
WFG3-2D	$0.002029_{0.001921}$	$0.002026_{0.001432}$	$0.001812_{0.001945}$	$0.001899_{0.001260}$
WFG1-3D	$0.104518_{0.008108}$	$0.359140_{0.159606}$	$0.104099_{0.007463}$	$0.285601_{0.097340}$
WFG3-3D	$0.004497_{0.002696}$	$0.004008_{0.001769}$	$0.003793_{0.002746}$	$0.003531_{0.001307}$
DTLZ7	$0.000758_{0.000010}$	$0.006579_{0.807318}$	$0.000758_{0.000012}$	$0.005407_{0.003795}$
DTLZ8	$0.000260_{0.000009}$	$0.000260_{0.000006}$	$0.000363_{0.000113}$	$0.000382_{0.000129}$

Table 2. GD. Median and interquartile range, for problems with $\gamma = 1.0$

	vNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vNSGA-II ($\mathcal{E}_{\mathcal{N}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{N}}$)
ZDT1	$0.000070_{0.000009}$	$0.000059_{0.000008}$	$0.000076_{0.000021}$	$0.000070_{0.000025}$
ZDT2	$0.002369_{0.000198}$	$0.002140_{0.000406}$	$0.000172_{0.000183}$	$0.000124_{0.000055}$
ZDT3	$0.000097_{0.000004}$	$0.000099_{0.000004}$	$0.000398_{0.003700}$	$0.000367_{0.000022}$
ZDT3-1	$0.000280_{0.000015}$	$0.000265_{0.000022}$	$0.000306_{0.000034}$	$0.000289_{0.000026}$
ZDT4	$0.000407_{0.000160}$	$0.000562_{0.000365}$	$0.000410_{0.000291}$	$0.000516_{0.000389}$
ZDT5	$0.412510_{0.010395}$	$0.449873_{0.018389}$	$0.414669_{0.012608}$	$0.448777_{0.014805}$
ZDT6	$0.000939_{0.000155}$	$0.000916_{0.000171}$	$0.000937_{0.000840}$	$0.000902_{0.000269}$
SZDT1	$0.001110_{0.000231}$	$0.000986_{0.000216}$	$0.001144_{0.000218}$	$0.001044_{0.000229}$
SZDT2	$0.001755_{0.000468}$	$0.001693_{0.000459}$	$0.001852_{0.000318}$	$0.001771_{0.000579}$
CTP1	$0.000078_{0.000011}$	$0.000077_{0.000009}$	$0.000089_{0.000110}$	$0.000059_{0.000066}$
CTP7	$0.000010_{0.000001}$	$0.000010_{0.000001}$	$0.000007_{0.000001}$	$0.000008_{0.000001}$
WFG1-2D	$0.100350_{0.005793}$	$0.122091_{0.027067}$	$0.087943_{0.003449}$	$0.096470_{0.011034}$
WFG3-2D	$0.002124_{0.001417}$	$0.001902_{0.000896}$	$0.002081_{0.001147}$	$0.001794_{0.001215}$
WFG1-3D	$0.121073_{0.003408}$	$0.122035_{0.004282}$	$0.125558_{0.005809}$	$0.125778_{0.004371}$
WFG3-3D	$0.029697_{0.001638}$	$0.029603_{0.001387}$	$0.030870_{0.001141}$	$0.031253_{0.001602}$
DTLZ7	$0.000939_{0.000020}$	$0.002613_{0.608852}$	$0.000943_{0.000019}$	$0.001891_{0.002317}$
DTLZ8	$0.000327_{0.000016}$	$0.000323_{0.000017}$	$0.000516_{0.000223}$	$0.000523_{0.000193}$

Table 3. IGD. Median and interquartile range, for problems with $\gamma = 0.5$

	vNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vNSGA-II ($\mathcal{E}_{\mathcal{N}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{N}}$)
ZDT1	$0.000070_{0.000007}$	$0.000075_{0.000009}$	$0.000068_{0.000003}$	$0.000070_{0.000005}$
ZDT2	$0.000153_{0.000011}$	$0.000149_{0.000015}$	$0.001287_{0.000464}$	$0.001131_{0.001280}$
ZDT3	$0.000241_{0.000192}$	$0.000563_{0.000218}$	$0.000488_{0.000156}$	$0.000444_{0.000219}$
ZDT3-1	$0.000122_{0.001940}$	$0.000305_{0.000119}$	$0.000136_{0.000046}$	$0.000230_{0.000081}$
ZDT4	$0.000363_{0.000316}$	$0.000424_{0.000377}$	$0.000305_{0.000287}$	$0.000394_{0.000339}$
ZDT5	$1.536616_{0.296067}$	$1.536616_{0.377491}$	$0.752388_{0.189020}$	$0.854478_{0.217997}$
ZDT6	$0.000224_{0.000054}$	$0.000404_{0.000076}$	$0.000189_{0.000032}$	$0.000303_{0.000061}$
SZDT1	$0.000486_{0.000119}$	$0.000639_{0.000183}$	$0.000433_{0.000098}$	$0.000574_{0.000135}$
SZDT2	$0.000705_{0.000140}$	$0.000911_{0.007692}$	$0.000584_{0.000117}$	$0.000889_{0.004394}$
CTP1	$0.000220_{0.000173}$	$0.000233_{0.000240}$	$0.000168_{0.000080}$	$0.000190_{0.000069}$
CTP7	$0.000019_{0.000001}$	$0.000019_{0.000001}$	$0.000009_{0.000000}$	$0.000009_{0.000000}$
WFG1-2D	$0.219719_{0.013864}$	$0.228637_{0.013330}$	$0.169609_{0.010101}$	$0.172915_{0.008610}$
WFG3-2D	$0.001742_{0.001718}$	$0.001742_{0.001330}$	$0.001483_{0.001562}$	$0.001593_{0.001153}$
WFG1-3D	$0.429655_{0.031484}$	$0.553228_{0.009443}$	$0.308903_{0.022632}$	$0.386909_{0.006128}$
WFG3-3D	$0.012801_{0.009288}$	$0.010835_{0.006396}$	$0.009525_{0.008510}$	$0.008529_{0.004455}$
DTLZ7	$0.004470_{0.000088}$	$0.007612_{1.400776}$	$0.004450_{0.000143}$	$0.006734_{0.003129}$
DTLZ8	$0.000485_{0.000108}$	$0.000524_{0.000105}$	$0.000401_{0.000026}$	$0.000407_{0.000040}$

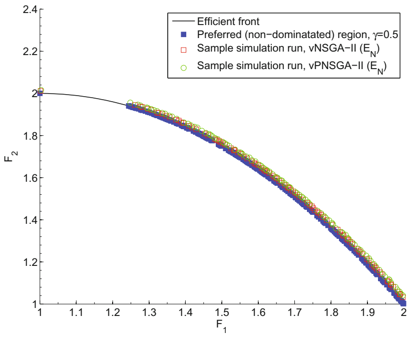


Fig. 3. $\mathcal{E}_{\mathcal{N}}$ and sample runs of vNSGA-II and vPNSGA-II on SZDT2

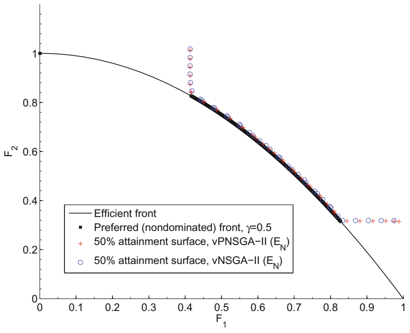


Fig. 4. $\mathcal{E}_{\mathcal{N}}$ and median attainment surface plot for ZDT2

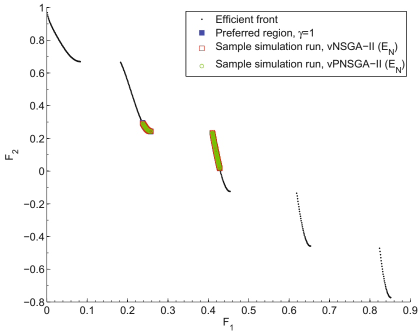


Fig. 5. $\mathcal{E}_{\mathcal{N}}$ and sample runs of vNSGA-II and vPNSGA-II on ZDT3

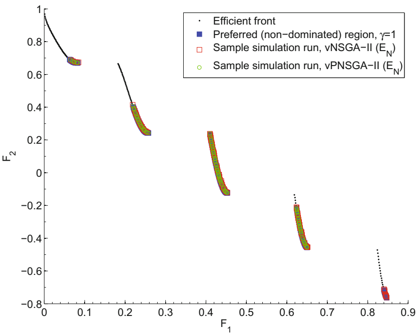


Fig. 6. $\mathcal{E}_{\mathcal{N}}$ and sample runs of vNSGA-II and vPNSGA-II on ZDT3-1

domination cone makes the original connected front disconnected (a connected part and the minimizer of F_1, see Figures 3 and 4). This is an additional difficulty that comes due to variable domination cones. However, the algorithms are able to find all these regions. Figures 5 to 12 show that the algorithms perform well on a large class of complex multi-objective problems with a variable domination structure.

Finally, Figures 13 and 14 nicely illustrate another characteristic of the variable domination structure. Both WFG3 and DTLZ1 have linear or planar efficient fronts. If we use guided-domination (or any polyhedral cone domination) based preference structures on these problems, we can only get a linear part of the efficient front. Any trade-off/ guided-domination based approach is polyhedral and is not suitable to focus on an arbitrary nonlinear subset of polyhedral efficient fronts. Variable domination on the other hand, with its inherent variability in the preference structure can however, focus on an arbitrary part of the efficient front of these problems (see Figure 14).

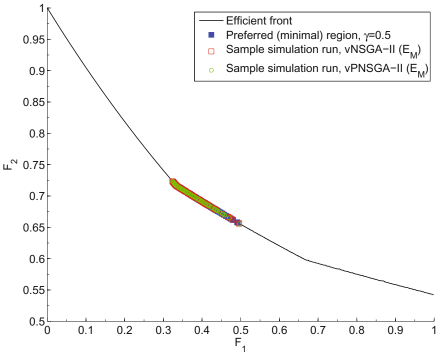


Fig. 7. $\mathcal{E}_{\mathcal{M}}$ and sample runs of vNSGA-II and vPNSGA-II on CTP1

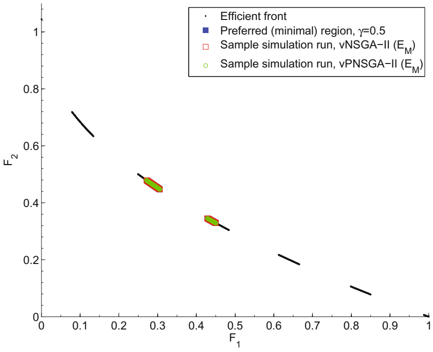


Fig. 8. $\mathcal{E}_{\mathcal{M}}$ and sample runs of vNSGA-II and vPNSGA-II on CTP7

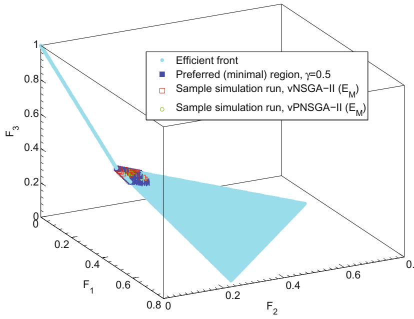


Fig. 9. $\mathcal{E}_{\mathcal{M}}$ and sample runs of vNSGA-II and vPNSGA-II on DTLZ8, with $\gamma = 0.5$

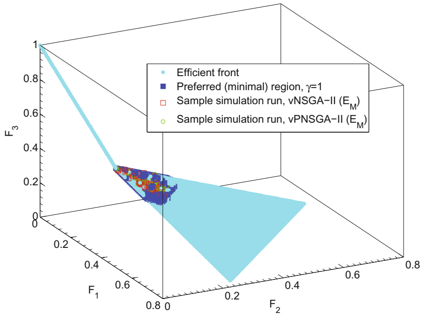


Fig. 10. $\mathcal{E}_{\mathcal{M}}$ and sample runs of vNSGA-II and vPNSGA-II on DTLZ8, with $\gamma = 1.0$

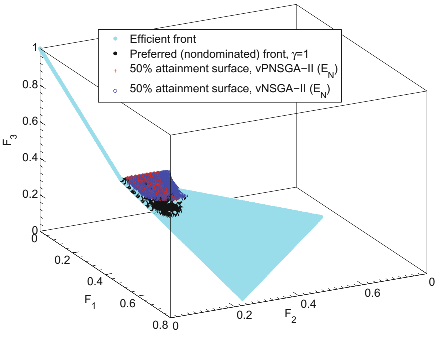


Fig. 11. $\mathcal{E}_{\mathcal{N}}$ and median attainment surface plot for DTLZ8

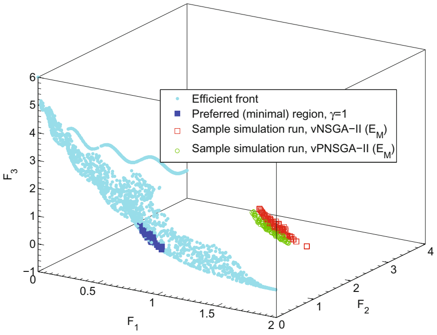


Fig. 12. $\mathcal{E}_{\mathcal{M}}$ and sample runs of vNSGA-II and vPNSGA-II on WFG1

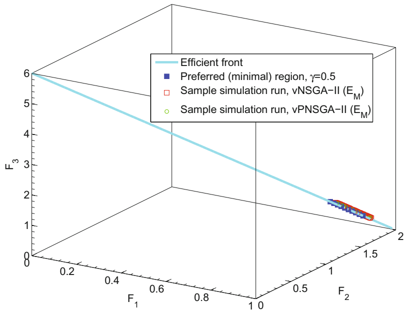


Fig. 13. $\mathcal{E}_{\mathcal{M}}$ and sample runs of vNSGA-II and vPNSGA-II on WFG3

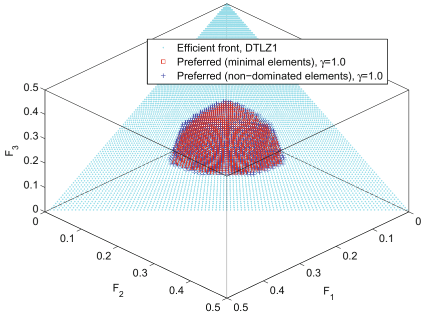


Fig. 14. $\mathcal{E}_{\mathcal{M}}$ and $\mathcal{E}_{\mathcal{N}}$ on DTLZ1

Table 4. IGD. Median and interquartile range, for problems with $\gamma = 1.0$

	vNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{M}}$)	vNSGA-II ($\mathcal{E}_{\mathcal{N}}$)	vPNSGA-II ($\mathcal{E}_{\mathcal{N}}$)
ZDT1	$0.000102_{0.000009}$	$0.000094_{0.000010}$	$0.000101_{0.000008}$	$0.000091_{0.000004}$
ZDT2	$0.000165_{0.000011}$	$0.000148_{0.000011}$	$0.000173_{0.000009}$	$0.000165_{0.000014}$
ZDT3	$0.000077_{0.000022}$	$0.000110_{0.000038}$	$0.000226_{0.000033}$	$0.000203_{0.000022}$
ZDT3-1	$0.000281_{0.000020}$	$0.000252_{0.000016}$	$0.000305_{0.000023}$	$0.000271_{0.000016}$
ZDT4	$0.000271_{0.000171}$	$0.000391_{0.000224}$	$0.000240_{0.000138}$	$0.000282_{0.000210}$
ZDT5	$0.955293_{0.113724}$	$1.069017_{0.209700}$	$0.752130_{0.138028}$	$0.731915_{0.153316}$
ZDT6	$0.000343_{0.000060}$	$0.000375_{0.000066}$	$0.001150_{0.000023}$	$0.001151_{0.000017}$
SZDT1	$0.000603_{0.000120}$	$0.000540_{0.000147}$	$0.000596_{0.000101}$	$0.000534_{0.000115}$
SZDT2	$0.000827_{0.000199}$	$0.000927_{0.010953}$	$0.000857_{0.000124}$	$0.000921_{0.001582}$
CTP1	$0.000084_{0.000007}$	$0.000082_{0.000008}$	$0.000052_{0.000011}$	$0.000052_{0.000014}$
CTP7	$0.000026_{0.000002}$	$0.000026_{0.000002}$	$0.000013_{0.000001}$	$0.000013_{0.000001}$
WFG1-2D	$0.112755_{0.006935}$	$0.116198_{0.007141}$	$0.065773_{0.004208}$	$0.069045_{0.004811}$
WFG3-2D	$0.001364_{0.000904}$	$0.001195_{0.000580}$	$0.001277_{0.000758}$	$0.001125_{0.000745}$
WFG1-3D	$0.189182_{0.005476}$	$0.189268_{0.005476}$	$0.153071_{0.004616}$	$0.151964_{0.003494}$
WFG3-3D	$0.020189_{0.007912}$	$0.020872_{0.005639}$	$0.018587_{0.005397}$	$0.018868_{0.005841}$
DTLZ7	$0.002302_{0.000255}$	$0.003606_{1.053387}$	$0.007218_{0.000075}$	$0.007680_{0.001558}$
DTLZ8	$0.000629_{0.000094}$	$0.000628_{0.000156}$	$0.000459_{0.000076}$	$0.000437_{0.000060}$

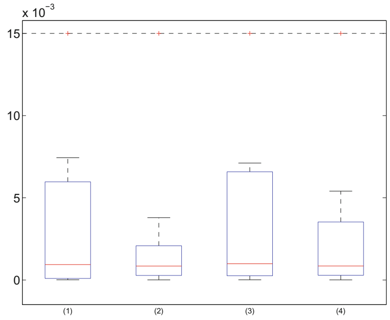


Fig. 15. Boxplot based on the GD values. (1)- vNSGA-II ($\mathcal{E}_{\mathcal{M}}$), (2)- vNSGA-II ($\mathcal{E}_{\mathcal{N}}$), (3)- vPNSGA-II ($\mathcal{E}_{\mathcal{M}}$), (4)- vPNSGA-II ($\mathcal{E}_{\mathcal{N}}$).

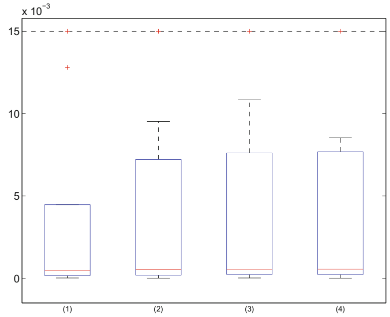


Fig. 16. Boxplot based on the IGD values. (1)- vNSGA-II ($\mathcal{E}_{\mathcal{M}}$), (2)- vNSGA-II ($\mathcal{E}_{\mathcal{N}}$), (3)- vPNSGA-II ($\mathcal{E}_{\mathcal{M}}$), (4)- vPNSGA-II ($\mathcal{E}_{\mathcal{N}}$).

4 Conclusions

This study considered multi-objective problems having a variable domination structure. It analyzed nondominated and minimal points and presented new theoretical results of an algorithmic value. Moreover, based on the theoretical results we presented two NSGA-II based algorithms, vNSGA-II and vPNSGA-II, to find a well-diverse set of both minimal and nondominated solutions. The test problems have adequately demonstrated that both these algorithms perform very well for finding optimal solutions even when the problem size and search space complexity is large. Although problems with a variable domination structure have many applications in mathematical, engineering and medical domain, this is the first study towards finding a well-diverse representation of minimal and nondominated solutions. This paper adequately shows that multi-objective evolutionary algorithms can deal well with the vagaries of a variable domination structure. More emphasis must be placed on developing interactive evolutionary multi-objective optimization techniques that incorporate variable domination structures. We hope that the study sheds adequate light in this area and motivates others.

References

[1] Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)

[2] Yu, P.L.: A class of solutions for group decision problems. Management Science 19, 936–946 (1973)

[3] Karasakal, E.K., Michalowski, W.: Incorporating wealth information into a multiple criteria decision making model. European J. Oper. Res. 150, 204–219 (2003)

[4] Eichfelder, G.: Vector optimization with a variable ordering structure. Technical Report 328, Preprint-Series of the Institute of Applied Mathematics, Univ. Erlangen-Nürnberg (2009)

[5] Wacker, M.: Multikriterielle optimierung bei registrierung medizinischer daten. Master's thesis, Uni. Erlangen Nürnberg (2008)

[6] Bergstresser, K., Yu, P.L.: Domination structures and multicriteria problems in n-person games. Theoy and Decision 8, 5–48 (1977)

[7] Ogryczak, W.: Inequality measures and equitable approaches to location problems. European Journal of Operational Research 122, 374–391 (2000)

[8] Ogryczak, W.: Inequality measures and equitable locations. Annals of Operations Research 167, 61–86 (2009)

[9] Ehrgott, M.: Multicriteria optimization, 2nd edn. Springer, Berlin (2005)

[10] Engau, A.: Variable preference modeling with ideal-symmetric convex cones. J. Global Optim. 42, 295–311 (2008)

[11] Shukla, P.K., Deb, K.: On finding multiple pareto-optimal solutions using classical and evolutionary generating methods. Eur. J. Oper. Res. 181, 1630–1652 (2007)

[12] Yu, P.L.: Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. J. Optim. Theory Appl. 14, 319–377 (1974)

[13] Huang, N.J., Rubinov, A.M., Yang, X.Q.: Vector optimization problems with nonconvex preferences. J. Global Optim. 40, 765–777 (2008)

[14] Rubinov, A.M., Gasimov, R.N.: Scalarization and nonlinear scalar duality for vector optimization with preferences that are not necessarily a pre-order relation. J. Global Optim. 29, 455–477 (2004)

[15] Yu, P.L.: Multiple-criteria decision making. Mathematical Concepts and Methods in Science and Engineering, vol. 30. Plenum Press, New York (1985)

[16] Wiecek, M.M.: Advances in cone-based preference modeling for decision making with multiple criteria. Decis. Mak. Manuf. Serv. 1, 153–173 (2007)

[17] Chen, G.Y.: Existence of solutions for a vector variational inequality: an extension of the Hartmann-Stampacchia theorem. J. Optim. Theory Appl. 74, 445–456 (1992)

[18] Chen, G.Y., Yang, X.Q.: Characterizations of variable domination structures via nonlinear scalarization. J. Optim. Theory Appl. 112, 97–110 (2002)

[19] Chen, G.Y., Huang, X., Yang, X.: Vector optimization. Lecture Notes in Economics and Mathematical Systems, vol. 541. Springer, Berlin (2005)

[20] Huang, N.J., Yang, X.Q., Chan, W.K.: Vector complementarity problems with a variable ordering relation. European J. Oper. Res. 176, 15–26 (2007)

[21] Sawaragi, Y., Nakayama, H., Tanino, T.: Theory of multiobjective optimization. Mathematics in Science and Engineering (1985)

[22] Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)

[23] Kung, H.T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. J. ACM 22, 469–476 (1975)

[24] Jensen, M.: Reducing the run-time complexity of multiobjective eas: The NSGA-II and other algorithms. IEEE Transactions on Evolutionary Computation 7, 503–515 (2003)

[25] Branke, J., Kauβler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. Advances in Engineering Software 32, 499–507 (2001)

[26] Shukla, P.K., Hirsch, C., Schmeck, H.: A framework for incorporating trade-off information using multi-objective evolutionary algorithms. In: PPSN (2), pp. 131–140 (2010)

[27] Deb, K., Pratap, A., Meyarivan, T.: Constrained test problems for multi-objective evolutionary optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 284–298. Springer, Heidelberg (2001)

[28] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 105–145. Springer, Heidelberg (2005)

[29] Huband, S., Hingston, P., Barone, L., White, L.: A review of multi-objective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10, 280–294 (2005)

[30] Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 584–593. Springer, Heidelberg (1996)

On the Computation of the Empirical Attainment Function

Carlos M. Fonseca[1,2,3], Andreia P. Guerreiro[4],
Manuel López-Ibáñez[5], and Luís Paquete[6]

[1] Department of Informatics Engineering, University of Coimbra,
Pólo II, 3030-290 Coimbra, Portugal
cmfonsec@dei.uc.pt
[2] CEG-IST, Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[3] DEEI, Faculty of Science and Technology, University of Algarve
Campus de Gambelas, 8005-139 Faro, Portugal
[4] Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
andreia.guerreiro@ist.utl.pt
[5] IRIDIA, Université Libre de Bruxelles (ULB)
Av. F. Roosevelt 50, CP 194/6, 1050 Brussels, Belgium
manuel.lopez-ibanez@ulb.ac.be
[6] CISUC, Department of Informatics Engineering, University of Coimbra,
Pólo II, 3030-290 Coimbra, Portugal
paquete@dei.uc.pt

Abstract. The attainment function provides a description of the location of the distribution of a random non-dominated point set. This function can be estimated from experimental data via its empirical counterpart, the empirical attainment function (EAF). However, computation of the EAF in more than two dimensions is a non-trivial task. In this article, the problem of computing the empirical attainment function is formalised, and upper and lower bounds on the corresponding number of output points are presented. In addition, efficient algorithms for the two and three-dimensional cases are proposed, and their time complexities are related to lower bounds derived for each case.

Keywords: Empirical attainment function, algorithms, multiobjective optimiser performance, estimation.

1 Introduction

The development of new stochastic optimisers, and comparison thereof, depends on the ability to assess their performance in some way. However, assessing the performance of stochastic *multiobjective* optimisers, such as multiobjective evolutionary algorithms, is considered difficult for two main reasons. On the one hand, theoretical convergence results are often unavailable for such optimisers, or are too weak to be indicative of their performance in practice. On the other

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 106–120, 2011.

hand, the analysis of experimental data is more challenging than in the single-objective case, due to the set nature of multiobjective optimisation outcomes.

Initial ideas on the performance assessment of stochastic multiobjective optimisers were put forward early in Evolutionary Multiobjective Optimisation history, based on the notion of attainment surfaces [2]. Those ideas were subsequently formalised in terms of the so-called *attainment function*, and links to existing results in random set theory were established [5]. As a mean-like, first-order moment measure of the statistical distribution of multiobjective optimisation outcomes, the attainment function provides a description of their *location* in objective space. More importantly, the attainment function may be estimated from experimental data using its empirical counterpart, the *empirical attainment function* (EAF). Thus, the performance of a stochastic multiobjective optimiser on a given optimisation problem, understood in terms of location of the corresponding outcome distribution, may be assessed by observing the outcomes of several independent optimisation runs and computing the corresponding EAF. Empirical comparisons may then be performed either visually or by formulating and testing appropriate statistical hypotheses.

Despite initial interest in the attainment function, much greater attention has been devoted in the literature to performance indicators, especially the hypervolume indicator [10], for which a body of theoretical and experimental results is currently available. In the meantime, the development of the attainment function approach has advanced slowly, and has focused mainly on theoretical aspects [4]. Due to computational difficulties, the EAF has been used mostly for visualisation purposes, in two dimensions [8].

In this article, the computation of the empirical attainment function is considered. The empirical attainment function and related concepts are introduced in Section 2, and the EAF computation problem is formalised in Section 3. In Section 4, bounds on the number of points to be computed are derived for two, three, and more dimensions. Efficient algorithms to compute the EAF in two and three dimensions are presented in Section 5, together with their computational complexities and some lower bounds on the complexity of the problem. The paper concludes with a discussion of the main contributions presented and directions for further work.

2 Background

When applied to an optimisation problem involving $d \geq 2$ objectives, stochastic multiobjective optimisers such as multiobjective evolutionary algorithms produce sets of solutions whose images in the d-dimensional objective space approximate the Pareto-optimal front of the problem in some sense. The quality of this approximation is usually considered to depend solely on the images in objective space, so that the outcome of an optimisation run may be seen as a set of points in $\mathbb{R}^d$. Typically, this is a subset of the *non-dominated*, or *minimal*, objective vectors evaluated during the run, since only such points effectively contribute to the quality of the approximation.

Definition 1 (Minima). *Given a set of points* $\mathrm{X} = \{x_1, \ldots, x_m \in \mathbb{R}^d\}$*, the set of minima of* X *under the component-wise order is the set*

$$\min \mathrm{X} = \{x \in \mathrm{X} : \forall y \in \mathrm{X}, y \leq x \Rightarrow y = x\} \tag{1}$$

Definition 2 (Non-dominated point set). *A set of points* X *such that* $\min \mathrm{X} = \mathrm{X}$ *is called a set of minima, or a non-dominated point set.*

In practice, the actual outcome sets produced for the same problem vary from optimisation run to optimisation run, due to the stochastic nature of the optimisers, and may be seen as *realisations* of a random non-dominated point set, or RNP set [3]. Optimiser performance may then be studied through the distribution of such a random set. In particular, the *attainment function* provides information about this distribution with respect to *location* [5,4], and is defined as the probability of an outcome set X attaining an arbitrary point $z \in \mathbb{R}^d$, i.e., the probability of $\exists x \in \mathrm{X} : x \leq z$. The symbol $\trianglelefteq$ is used to denote attainment of a point by a set, e.g., $\mathrm{X} \trianglelefteq z$.

The attainment function may be estimated from experimental data through its empirical version:

Definition 3 (Empirical attainment function). *Let* $\mathbf{I}\{\cdot\} : \mathbb{R}^d \longmapsto \{0, 1\}$ *denote the indicator function, and let* $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$ *be non-dominated point set realisations drawn independently from some RNP set distribution. The empirical attainment function (EAF) is the discrete function* $\alpha_n : \mathbb{R}^d \longmapsto [0, 1]$*, where*

$$\alpha_n(z) = \alpha_n(\mathrm{X}_1, \ldots, \mathrm{X}_n; z) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}\{\mathrm{X}_i \trianglelefteq z\} \tag{2}$$

This definition clearly describes how to evaluate the EAF at a given point $z \in \mathbb{R}^d$, but, in practice, it is also necessary to decide at which points the EAF should be evaluated. For visualisation purposes [8], for example, the boundaries of the regions of the objective space where the EAF takes a constant value are usually of interest. These boundaries were originally referred to as *%-attainment surfaces* [2], and may be understood as the family of tightest goal vectors individually attained by a given percentage of the optimisation runs considered. They have also been described as summary attainment surfaces [6]. Visualisation of the EAF, and of differences between EAFs associated to different optimisers, is often used as an exploratory data analysis tool to obtain additional insight into optimiser performance [6,8].

To produce a graphical representation of the EAF, the desired summary attainment surfaces must be computed in some way. The above description suggests that a summary attainment surface may be represented by a set of minima, and this set can be easily computed in two dimensions as discussed later in Section 5. Yet, no formal definition of the corresponding computation problem, or algorithm to solve it in more than two dimensions, have been presented to date. To overcome this difficulty, Knowles [6] proposed a plotting method based on the computation of the intersections between a number of axis-aligned sampling lines and the summary attainment surface of interest. These intersections provide a grid-like sampling of the surface which is fast to compute, and leads to plots that are easy to interpret, at least in three dimensions. However, this sampling provides only an approximate description of the desired attainment surface while potentially involving a much larger number of points than the exact description adopted in this work.

3 The EAF Computation Problem

Summary attainment surfaces, as described in the previous section, may be understood as the lower boundary of the corresponding EAF superlevel sets. Formally, denote the t/n-superlevel set of $\alpha_n(z)$, $t = 1, \dots, n$, by:

$$\mathrm{V}_{t/n} = \{z \in \mathbb{R}^d : \alpha_n(z) \geq t/n\} \tag{3}$$

and the corresponding set of minima, which Proposition 1 will show to be finite although $\mathrm{V}_{t/n}$ is infinite, by $\mathrm{L}_t = \min \mathrm{V}_{t/n}$. Since $\alpha_n(z)$ is a coordinate-wise non-decreasing function, $\mathrm{V}_{t/n}$ is equal to the upper set of L_t, i.e.:

$$\mathrm{V}_{t/n} = \{z \in \mathbb{R}^d : \mathrm{L}_t \trianglelefteq z\} \tag{4}$$

Thus, the EAF computation problem may be defined as:

Problem 1 (EAF computation). Given an input sequence of non-empty non-dominated point sets:

$$\mathrm{S} = (\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) \tag{5}$$

containing

$$m = \sum_{i=1}^{n} m_i, \quad m_i = |\mathrm{X}_i| \tag{6}$$

input points, find the output sequence

$$\mathrm{R} = (\mathrm{L}_1, \mathrm{L}_2, \dots, \mathrm{L}_n) \tag{7}$$

where L_t, $t = 1, \dots, n$, denotes the set of minima of the t/n-superlevel set, $\mathrm{V}_{t/n}$, of $\alpha_n(\mathrm{X}_1, \dots, \mathrm{X}_n; z)$. The total number of output points is

$$\ell = \sum_{t=1}^{n} \ell_t, \quad \ell_t = |\mathrm{L}_t| \tag{8}$$

It remains to show how the output sets $\mathrm{L}_1, \dots, \mathrm{L}_n$ relate to the input sequence, S. For each $t = 1, \dots, n$, the auxiliary set

$$\mathrm{J}_t = \left\{ \bigvee_{i=1}^{t} z_i : (z_1, \dots, z_t) \in \prod_{i=1}^{t} \mathrm{X}_{j_i},\ j_i \in \{1, \dots, n\} \wedge (a < b \Leftrightarrow j_a < j_b) \right\} \tag{9}$$

is defined, where $\bigvee_{i=1}^{t} z_i$ denotes the *join* (component-wise maximum, or least upper bound) of points $z_1, \dots, z_t \in \mathbb{R}^d$, $\prod_{i=1}^{t} \mathrm{X}_{j_i}$ denotes the Cartesian product of sets $\mathrm{X}_{j_1}, \dots, \mathrm{X}_{j_t}$, and $(\mathrm{X}_{j_1}, \dots, \mathrm{X}_{j_t})$ is any length-t subsequence of S. Then, the following holds true:

Proposition 1. *L_t is finite and equal to the set of minima of J_t, $t = 1, \ldots, n$.*

Proof.

1. $\mathrm{J}_t \subseteq \mathrm{V}_{t/n}$, since all elements of J_t are attained by at least t distinct input sets by construction.
2. Each minimum of $\mathrm{V}_{t/n}$ must be the least upper bound (join) of t points from distinct input sets, which is in J_t by construction. Therefore, $\min \mathrm{V}_{t/n} \subseteq \mathrm{J}_t$.
3. Together, 1. and 2. imply that $\min \mathrm{V}_{t/n} \subseteq \min \mathrm{J}_t$.
4. $\min \mathrm{J}_t \subseteq \min \mathrm{V}_{t/n}$. Assume that a minimum of J_t is not a minimum of $\mathrm{V}_{t/n}$. Then, there must be a minimum of $\mathrm{V}_{t/n}$ which dominates it. Given 3., this minimum of $\mathrm{V}_{t/n}$ must be in J_t as well, which gives rise to a contradiction.
5. Together, 3. and 4. imply that $\min \mathrm{V}_{t/n} = \min \mathrm{J}_t$. Since J_t is finite, so is L_t.

4 The Size of the Output Sets

As the complexity of any algorithm for the EAF computation problem will necessarily be bounded below by the size of the output sets $\mathrm{L}_1, \ldots, \mathrm{L}_n$, the maximum size of these sets is of interest.

4.1 The Two-Objective Case

When $d = 2$, an upper bound for the total number of output points, ℓ, can be easily obtained by noting that all output sets L_t, $t = 1, \ldots, n$, are non-dominated point sets, which implies that all points in L_t must have distinct coordinates. Therefore, the cardinality $\ell_t = |\mathrm{L}_t|$ is bounded above by the total number of input points, m. In other words, $\ell_t \in O(m)$, which leads to $\ell \in O(nm)$.

Furthermore, the above bound can be shown to be tight by means of an example. Given two positive integers n and m such that $m \geq n$, consider the input sequence:

$$\mathrm{S} = (\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n) \tag{10}$$

with

$$\mathrm{X}_i = \big\{(j, m - j + 1) : (j - i) \bmod n = 0,\ j \in \{1, \ldots, m\}\big\} \tag{11}$$

for all $i = 1, \ldots, n$. Then, $\ell_t = m - t + 1$ and $\ell = n(2m - n + 1)/2$. Since $m \geq n$, $\ell \in \Omega(nm)$, and the following proposition holds true:

Proposition 2. *In the 2-dimensional EAF computation problem, $\ell \in \Theta(nm)$.*

4.2 The Three-Objective Case

To establish an upper bound on the maximum total number of output points when $d = 3$, assume, without loss of generality, that all input points have distinct coordinates.[1] Recall that output points are the join of points from different input

[1] If this is not the case, add sufficiently small perturbations to the common coordinate values in the input so that all coordinates become distinct. Note that the number of minimal elements of J_t, $t = 1, \ldots, n$, may increase as a result, but cannot decrease.

sets, and that all output sets L_t, $t = 1, \ldots, n$, are non-dominated point sets by definition. In particular, L_1 is the set of minima of $\mathrm{J}_1 = \bigcup_{i=1}^{n} \mathrm{X}_i$, which implies that $\ell_1 \in O(m)$.

Since the number of objectives is $d = 3$, points in each output set L_t must differ from each other in at least two coordinates, or one would dominate the other. Thus, for each input point $p = (p^x, p^y, p^z) \in \mathrm{J}_1$, there may be at most two distinct points $q = (q^x, q^y, q^z)$ in each output set L_t such that $(q^x, q^z) = (p^x, p^z)$ and $q^y \geq p^y$, or $(q^y, q^z) = (p^y, p^z)$ and $q^x \geq p^x$. Note that this does not exclude the case where $q = p$, nor does it imply that such a point $q \in \mathrm{L}_t$ exists.

Moreover, for each point $q = (p^x, q^y, p^z) \in \mathrm{L}_t$ considered above, there may be at most one point $r = (r^x, r^y, r^z)$ in each L_j, $t < j \leq n$, such that $(r^x, r^y) = (p^x, q^y)$ and $r^z > p^z$, and similarly for each $q = (q^x, p^y, p^z) \in \mathrm{L}_t$. Thus, each input point p may be associated with $O(n)$ output points of the form q above and with $O(n^2)$ output points of the form r.

Finally, every output point must be of one of these two types, q or r. This is clearly true for every output point that is either an input point as well, or is dominated by some input point while differing from it only in the value of a single, x or y, coordinate, which corresponds to type q. In all other cases, the join of the input point(s) defining the x and y coordinates of a given output point will differ from that output point in the z coordinate value only, which must now be equal to that of (one of) the other input point(s) considered. This join must be a minimum of J_j for some $j < t$, or the original output point would not be a minimum of J_t, either. Therefore, type r applies.

Since there are m input points in total, the following holds true:

Proposition 3. *In the 3-dimensional EAF computation problem, $\ell \in O\left(n^2 m\right)$.*

4.3 More Than Three Objectives

As shown above, the maximum number of output points, ℓ, grows at most linearly with the total number of input points, m, when the number of objectives is two or three, and the number of input sets, n, is constant. Unfortunately, this result does not carry over to more than three objectives. As an example, consider, for given positive integers m_1 and m_2, that:

$$\mathrm{S} = (\mathrm{X}_1, \mathrm{X}_2) \tag{12}$$

$$\mathrm{X}_1 = \{(j_1, m_1 - j_1 + 1, 0, 0) : j_1 = 1, \ldots, m_1\} \tag{13}$$

$$\mathrm{X}_2 = \{(0, 0, j_2, m_2 - j_2 + 1) : j_2 = 1, \ldots, m_2\} \tag{14}$$

Then,

$$\mathrm{J}_1 = \mathrm{X}_1 \cup \mathrm{X}_2 \tag{15}$$

$$\mathrm{J}_2 = \{(j_1, m_1 - j_1 + 1, j_2, m_2 - j_2 + 1) : j_1 = 1, \ldots, m_1,\ j_2 = 1, \ldots, m_2\} \tag{16}$$

Since, in this case, both J_1 and J_2 are non-dominated point sets, $(\mathrm{L}_1, \mathrm{L}_2) = (\mathrm{J}_1, \mathrm{J}_2)$, and

$$|\mathrm{L}_1| = m_1 + m_2 = m \tag{17}$$

$$|\mathrm{L}_2| = m_1 m_2 \tag{18}$$

Algorithm 1. EAF computation

```
for $t = 1$ to $n$ do
    compute $\mathrm{J}_t$ from $\mathrm{X}_1, \ldots, \mathrm{X}_n$
    $\mathrm{L}_t \leftarrow \mathsf{minima}(\mathrm{J}_t)$
```

Setting $m_1 = m_2 = m/2$, the total number of output points is $\ell = m + m^2/4$, which establishes a lower bound of $\Omega(m^2)$ in the four-objective case.

The above lower bound can be shown to apply also to higher numbers of objectives (through dimension embedding), but tighter bounds for increasing values of d can be achieved by extending the proposed construction. Assuming that d is constant and even, consider $n = d/2$ input sets:

$$\mathrm{X}_1 = \{(j_1, m_1 - j_1 + 1, 0, 0, \ldots, 0, 0) \in \mathbb{R}^d : j_1 = 1, \ldots, m_1\} \tag{19}$$

$$\mathrm{X}_2 = \{(0, 0, j_2, m_2 - j_2 + 1, 0, 0, \ldots, 0, 0) \in \mathbb{R}^d : j_2 = 1, \ldots, m_2\} \tag{20}$$

$$\vdots$$

$$\mathrm{X}_i = \{(0, 0, \ldots, 0, 0, j_i, m_i - j_i + 1, 0, 0, \ldots, 0, 0) \in \mathbb{R}^d : j_i = 1, \ldots, m_i\} \tag{21}$$

$$\vdots$$

$$\mathrm{X}_n = \{(0, 0, \ldots, 0, 0, j_n, m_n - j_n + 1) \in \mathbb{R}^d : j_n = 1, \ldots, m_n\} \tag{22}$$

of equal size $m_1 = \cdots = m_n = m/n$, and focus on the cardinality of $|\mathrm{L}_n| = |\mathrm{J}_n| = (m/n)^n = (m/n)^{d/2}$. Then, the following general result may be stated:

Proposition 4. *In the d-dimensional EAF computation problem, the maximum total number of output points is* $\Omega\left(m^{\lfloor d/2 \rfloor}\right)$.

5 Time Complexity and Algorithms

The cardinality lower bounds derived in the previous section provide trivial lower bounds on the time complexity of the EAF computation problem. Additionally, the known lower bound of $O(n \log n)$ on the complexity of finding the minima (or, alternatively, the maxima) of a point set [9, Theorem 4.8] applies to EAF computation as well, since $\mathrm{L}_1 = \min \mathrm{J}_1$. Formally:

Proposition 5. *In the comparison-tree model, any algorithm that solves the EAF computation problem in d dimensions requires time* $\Omega\left(m \log m + m^{\lfloor d/2 \rfloor}\right)$. *The time required when* $d = 2$ *is* $\Omega\left(m \log m + nm\right)$.

The design of algorithms for the EAF computation problem is approached here by noting that $\mathrm{L}_t = \min \mathrm{J}_t$, as stated in Proposition 1, which immediately suggests dividing the problem into two main computation steps, as outlined in Algorithm 1. The disadvantage of such a brute-force approach is that $|\mathrm{J}_t|$ grows exponentially in t, leading to overall exponential runtime growth in n, even in two or three dimensions.

Algorithm 2. Minima of a set of points

```
Input: X // a set of points in $\mathbb{R}^d$
Output: $L_1$ // the set of minima of X
 1: $m \leftarrow |X|$
 2: Q is a queue containing X sorted in *ascending* order of coordinate $d$
 3: $L_1 \leftarrow \varnothing$
 4: $L_1^* \leftarrow \varnothing$
 5: while $Q \neq \varnothing$ do
 6:    $p \leftarrow \mathsf{pop}(Q)$
 7:    $p^*$ is the projection of $p$ onto the first $d-1$ dimensions
 8:    if $L_1^* \ntrianglelefteq p^*$ then
 9:       $L_1^* \leftarrow \mathsf{minima}(L_1^* \cup \{p^*\})$
10:       $L_1 \leftarrow L_1 \cup \{p\}$
11: return $L_1$
```

A better alternative consists of alternating between J_t computation steps and L_t computation steps, while avoiding generating points in J_t which would be dominated by those already in L_t. Such an approach is consistent with the well-known dimension-sweep paradigm [9, p. 10f] of computational geometry, and the EAF algorithms developed in this work are based on existing dimension-sweep algorithms for minima [7].

Consider the computation of L_1, which, as pointed out earlier, consists of determining the minima of all input points, regardless of the input set to which each point actually belongs. A general solution to this problem [9, p. 160] is outlined in Algorithm 2, under the common assumption that all input points are distinct, and have distinct coordinate values. The algorithm starts from an empty output set L_1, and visits input points in ascending order of their last coordinate, i.e., it sweeps X along the last dimension. Clearly, a newly visited point cannot dominate previously visited points, but it will be dominated by earlier points whenever this is true with respect to the first $d-1$ coordinates. Therefore, it suffices to check the projected point p^* against a set, L_1^*, of minimal projections in order to decide whether or not p itself is a minimum. Due to this dimensionality reduction, efficient dimension-sweep algorithms can be obtained for the minima problem in two and three dimensions by specialising the dominance check and update steps (lines 8–9) in each case.

In practice, input coordinate values may not be all distinct, and adjustments to dimension-sweep algorithms in their basic form may be required, which can often be made without compromising the worst-case complexity of the original versions of the algorithms. In the following, no distinct-coordinate assumptions are made, and repeated coordinate values and/or non-disjoint input sets, $X_1, \ldots, X_n$, are explicitly allowed in the input to the EAF algorithms presented.

5.1 The Two-Objective Case

The general approach of Algorithm 2 becomes particularly simple when $d = 2$. In that case, p^* is a scalar, and L_1^* may contain at most one element. Therefore,

Algorithm 3. EAF computation in two dimensions

```
**Input:** $\mathrm{S} = (\mathrm{X}_1, \ldots, \mathrm{X}_n)$ // a sequence of non-dominated point sets
**Output:** $\mathrm{R} = (\mathrm{L}_1, \ldots, \mathrm{L}_n)$ // a sequence of non-dominated point sets
 1: $\mathrm{X} \leftarrow \biguplus_{i=1}^{n} \mathrm{X}_i$ // multiset sum, duplicate points are allowed
 2: $m \leftarrow \sum_{i=1}^{n} |\mathrm{X}_i|$
 3: $\mathrm{X}^x$ is X sorted in *descending* order of the $x$ coordinate
 4: $\mathrm{X}^y$ is X sorted in *ascending* order of the $y$ coordinate
 5: **for** $t = 1$ **to** $n$ **do**
 6:    $\mathrm{L}_t \leftarrow \varnothing$
 7:    $\mathrm{Q}^x \leftarrow \mathrm{X}^x$ // initialise queue
 8:    $\mathrm{Q}^y \leftarrow \mathrm{X}^y$ // initialise queue
 9:    $\mathrm{A} \leftarrow \varnothing$
10:    $\mathit{level} \leftarrow 0$
11:    $p \leftarrow (\infty, -\infty)$
12:    **while** $\mathrm{Q}^x \neq \varnothing$ **and** $\mathrm{Q}^y \neq \varnothing$ **do**
13:       **repeat**
14:          $q \leftarrow \mathsf{pop}(\mathrm{Q}^y)$
15:          **if** $q^x < p^x$ **then**
16:             **if** $\mathsf{input\_set}(q) \notin \mathrm{A}$ **then**
17:                $\mathit{level} \leftarrow \mathit{level} + 1$
18:             $\mathrm{A} \leftarrow \mathrm{A} \uplus \{\mathsf{input\_set}(q)\}$ // multiset sum
19:       **until** $\mathrm{Q}^y = \varnothing$ **or** $(\mathit{level} \geq t\ \underline{\textbf{and}\ q^y \neq \mathsf{top}(\mathrm{Q}^y)^y})$
20:       **if** $\mathit{level} \geq t$ **then**
21:          **repeat**
22:             $p \leftarrow \mathsf{pop}(\mathrm{Q}^x)$
23:             **if** $p^y \leq q^y$ **then** // $(p^x, q^y) \in \mathrm{J}_t$
24:                $\mathrm{A} \leftarrow \mathrm{A} \setminus \{\mathsf{input\_set}(p)\}$ // multiset difference
25:                **if** $\mathsf{input\_set}(p) \notin \mathrm{A}$ **then**
26:                   $\mathit{level} \leftarrow \mathit{level} - 1$
27:          **until** $\mathit{level} < t\ \underline{\textbf{and}\ (\mathrm{Q}^x = \varnothing\ \textbf{or}\ p^x \neq \mathsf{top}(\mathrm{Q}^x)^x)}$
28:          $\mathrm{L}_t \leftarrow \mathrm{L}_t \cup \{(p^x, q^y)\}$
29: **return** $(\mathrm{L}_1, \ldots, \mathrm{L}_n)$
```

each iteration of the while loop can be performed in $O(1)$ time, and the algorithm runs in asymptotically optimal, $O(m \log m)$ time due to sorting in line 2.

Extending this approach to the full EAF computation problem in two dimensions is not only possible, but a C implementation of such an algorithm was contributed by the first author to the PISA platform [1] in 2005. A somewhat simpler, but functionally-equivalent, algorithm to compute the EAF in two dimensions is presented as Algorithm 3.

Input sets are merged, pre-sorted according to each coordinate, and stored into two queues, Q^x and Q^y. The following operations can be performed in constant time: $\mathsf{top}(\mathrm{Q})$ returns the element at the top of a queue Q; $\mathsf{pop}(\mathrm{Q})$ retrieves the element at the top and removes it from Q; $\mathsf{input\_set}(p)$ returns the index of the input set containing p.

Each output set L_t is computed independently from the others. For each $t = 1, \ldots, n$, and starting from an empty L_t, input points are visited in ascending order of their y coordinates until points from at least t different input sets have been visited (lines 13–19). The last point visited, q, establishes the smallest value of the y coordinate of any point in J_t and, thus, of any of its minima. A second sweep is then made in descending order of x-coordinate values (lines 21–27). For each point p thus visited, if it is such that $p^y \leq q^y$, then (p^x, q^y) must be an element of J_t. The number of points from each input set which dominate (p^x, q^y) is tracked using multiset A and the variable *level*.

Note that, as long as the second repeat-until loop has not exited, the number of input sets that attain (p^x, q^y) must be at least t. Also, p and q must be either the same point or belong to distinct input sets. Otherwise, the first repeat-until loop would have exited before q was reached. In addition, if the current point p is the only one from its input set to dominate (p^x, q^y), then (p^x, q^y) is actually a minimum of J_t and, consequently, an element of L_t.

This process is iterated by alternating between sweeps along the y and x dimensions until either queue is exhausted. A new minimum of J_t is obtained after each iteration of the outer while loop (line 12–28), with the possible exception of the last one. As a result, the exponential growth associated with the full enumeration of J_t is avoided because, once a minimum of J_t is found, no further elements of J_t dominated by that minimum are ever generated. Repeated coordinate values are handled by the conditions underlined in lines 19 and 27.

Regarding algorithmic complexity, sorting the input requires $O(m \log m)$ time, and each output set L_t is generated in $O(m)$ time. Since the number of output sets is n, the overall time complexity of the algorithm is $O(m \log m + nm)$, which matches the corresponding lower bound stated in Proposition 5. Algorithm 3 is, therefore, asymptotically optimal.

5.2 The Three-Objective Case

Asymptotically optimal algorithms for minima in three dimensions can also be obtained from Algorithm 2. Since L_1^* is now a set of minima in two dimensions, it admits a total order, and may be organised as a height-balanced binary search tree on either of the first two coordinates. This allows the dominance check in line 8 to be performed in $O(\log m)$ time and the L_1^* update in line 9 to be performed in amortised $O(\log m)$ time. Indeed, each update consists of a search operation and at most one insertion, both of which have complexity $O(\log m)$ time, plus a variable number of removals[2] which, in total, cannot exceed m. Since each removal can also be performed in $O(\log m)$ time, the overall complexity is $O(m \log m)$, and the algorithm is asymptotically optimal [7].

Extending this approach to the EAF computation problem in three dimensions is much less straightforward than it was in the two-dimensional case. A discussion of the main aspects of the solution proposed as Algorithm 4 follows.

Data structures. Since all input sets $\mathrm{X}_1, \ldots, \mathrm{X}_n$ and output sets $\mathrm{L}_1, \ldots, \mathrm{L}_n$ are non-dominated point sets, $2n$ data structures based on a height-balanced binary search tree, as in the minima algorithm described above, are used to manage

[2] Each point to be removed may be found in constant time if the tree is threaded.

Algorithm 4. EAF computation in three dimensions

```
Input: $S = (X_1, \ldots, X_n)$ // a sequence of non-dominated point sets
Output: $R = (L_1, \ldots, L_n)$ // a sequence of non-dominated point sets
$X = \biguplus_{i=1}^{n} X_i$ // multiset sum, duplicate points are allowed
$m \leftarrow \sum_{i=1}^{n} |X_i|$
Q is X sorted in *ascending* order of the $z$ coordinate
$L_t \leftarrow \varnothing$, $t = 1, \ldots, n$
$L_t^* \leftarrow \{(-\infty, \infty, -\infty), (\infty, -\infty, -\infty)\}$, $t = 1, \ldots, n$ // Sentinels
$X_i^* \leftarrow \{(-\infty, \infty, -\infty), (\infty, -\infty, -\infty)\}$, $i = 1, \ldots, n$ // Sentinels
$p \leftarrow \mathsf{pop}(Q)$
$j \leftarrow \mathsf{input\_set}(p)$
$\mathsf{insert}(p, X_j^*)$
$\mathsf{insert}(p, L_1^*)$
$A \leftarrow \{j\}$
$t_{\max} \leftarrow 1$
while $Q \neq \varnothing$ do
    $p \leftarrow \mathsf{pop}(Q)$
    $j \leftarrow \mathsf{input\_set}(p)$
    $q \leftarrow \mathsf{floor}^x(p, X_j^*)$
    if $p^y < q^y$ then // always true if $X_j$ is indeed a non-dominated point set
        $t \leftarrow t_{\max}$
        $t_{\min} \leftarrow 1$
        while $t \geq t_{\min}$ do
            $r \leftarrow \mathsf{floor}^x(p, L_t^*)$
            if $r^y \leq p^y$ then
                $t_{\min} \leftarrow t + 1$
            else if $r^y < q^y$ then
                $s_t \leftarrow (p^x, r^y, p^z)$
            else
                $s_t \leftarrow \mathsf{lower}^y(q, L_t^*)$
            $t \leftarrow t - 1$
        repeat
            $q \leftarrow \mathsf{higher}^x(q, X_j^*)$
            $b \leftarrow \max(p^y, q^y)$
            for $t = t_{\max}$ down to $t_{\min}$ do
                while $s_t^y \geq b$ and $(s_t^y > b$ or $b > p^y)$ do
                    if $s_t^x \geq q^x$ then
                        $s_t \leftarrow \mathsf{lower}^y(q, L_t^*)$
                    else
                        submit $(s_t^x, s_t^y, p^z)$ to $L_{t+1}^*$
                        $s_t \leftarrow \mathsf{higher}^x(s_t, L_t^*)$
        until $q^y \leq p^y$
        for $t = t_{\max}$ down to $t_{\min}$ do
            if $s_t^x < q^x$ then
                submit $(s_t^x, p^y, p^z)$ to $L_{t+1}^*$
        submit $p$ to $X_j^*$
        submit $p$ to $L_{t_{\min}}^*$
    if $j \notin A$ then
        $A \leftarrow A \cup \{j\}$
        $t_{\max} \leftarrow \min(t_{\max} + 1, n - 1)$
$L_t \leftarrow L_t \cup (L_t^* \setminus \{(-\infty, \infty, -\infty), (\infty, -\infty, -\infty)\})$, $t = 1, \ldots, n$
return $(L_1, \ldots, L_n)$
```

them. Points from each set are organised in the corresponding data structure with respect to their projection onto the xy-plane, which conveniently allows both x and y coordinates to be used as search key as long as the projections of all points stored are non-dominated. Insertion, removal and the following search operations can be performed in logarithmic time on such a data structure, X^*:

floor$^x(p, \mathrm{X}^*)$ The point $q \in \mathrm{X}^*$ with the greatest $q^x \leq p^x$
lower$^x(p, \mathrm{X}^*)$ The point $q \in \mathrm{X}^*$ with the greatest $q^x < p^x$
ceiling$^x(p, \mathrm{X}^*)$ The point $q \in \mathrm{X}^*$ with the least $q^x \geq p^x$
higher$^x(p, \mathrm{X}^*)$ The point $q \in \mathrm{X}^*$ with the least $q^x > p^x$

The corresponding operations with respect to the y coordinate are available as well, and have the same complexity.

Algorithm description. Output sets are computed by sweeping along the z dimension and searching for the different types of output points identified in Section 4. As a consequence, all output sets are computed concurrently, instead of independently from each other.

For simplicity, begin by assuming that no repeated z-coordinate values appear in the input, a restriction which will be lifted later. Before entering the main loop in line 13, the input data is queued in ascending order of the z coordinate, and individual data structures X_i^* and L_t^*, $i, t = 1, \ldots, n$, are created and initialised with sentinels (lines 1–6), so that all search operations are guaranteed to return a point. Then, the first point retrieved from the queue is inserted into the corresponding X_j^* and into L_1^* (lines 7–10), as it must be a minimal element of J_1. Set A is used to track which input sets have been visited so far.

In the main loop, each new input point dequeued is checked to ascertain that it is not dominated by other points in its input set (lines 14–17). Then, for each non-empty $\mathrm{L}_{n-1}, \mathrm{L}_{n-2}, \ldots, \mathrm{L}_1$, new output points are generated as follows:

1. For each input point $p \in \mathrm{X}$, element of some input set X_j, $j \in \{1, \ldots, n\}$, an output point $r = (r^x, r^y, r^z) \in \mathrm{L}_t$ is sought, such that $\mathrm{X}_j \ntrianglelefteq r$, $(p^x, p^z) \geq (r^x, r^z)$ and $p^y < r^y$. This is depicted as point 1 in Fig. 1. Then, $s = (p^x, r^y, p^z)$ must be an element of J_{t+1}, as it is attained by one more input set than r. In addition, if s^y is required to be minimal, s will be an element of L_{t+1}. Such points, r, if they exist, are identified in the first inner loop of the algorithm (lines 20–28), and the corresponding output point, s, depicted as point A in the figure, is generated in line 25. For convenience, it is output only later, in line 37.
2. For each input point $p \in \mathrm{X}$, element of some input set X_j, $j \in \{1, \ldots, n\}$, all output points $s = (s^x, s^y, s^z) \in \mathrm{L}_t$ such that $\mathrm{X}_j \ntrianglelefteq s$, $(p^x, p^y) < (s^x, s^y)$ and $p^z \geq s^z$ are sought. Then, (s^x, s^y, p^z) must also be an element of J_{t+1}. Since input points are processed in ascending order of z-coordinate values, (s^x, s^y, p^z) will be an element of L_{t+1}. Such points, if they exist, are determined in the second inner-loop (lines 29–39) and, eventually, also as the last point found in the first inner loop, in line 27. They are depicted as points 2, 3 and 6 in Fig. 1.

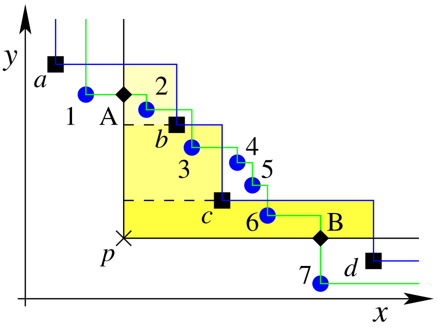


Fig. 1. Example, where $L_t^* = \{1, ..., 7\}$, $X_j^* = \{a, b, c, d\}$, and p is the new point in X_j

Algorithm 5. Submit u to L_t^*

```
1: $v \leftarrow \mathsf{floor}^x(u, L_t^*)$
2: if $u^y < v^y$ then
3:    for all $w \in L_t^* : (u^x, u^y) \leq (w^x, w^y)$ do
4:       if $u^z > w^z$ then
5:          $L_t \leftarrow L_t \cup \{w\}$
6:       $\mathsf{remove}(w, L_t^*)$
7:    $\mathsf{insert}(u, L_t^*)$
```

3. In the third inner loop (lines 40–42), output points analogous to those determined in the first loop, but with the roles of the x and y coordinates reversed, are computed (point B in the figure).
4. Finally, each input point p will itself be a member of the output set $L_{t_{\min}}$, where $t_{\min}$ is the lowest index t such that L_t^* does not attain p in the current iteration of the outer loop. The value of $t_{\min}$ is determined in lines 22–23, and L_t is updated in line 44.

The L_t^* data structures are updated as detailed in Algorithm 5. Provided that the new point u is not dominated by the points currently in L_t^* (line 2), points w in L_t^* whose projections are dominated by u are removed (lines 3–6), and the new point u is inserted (line 7). Otherwise, u is simply discarded.

Repeated z-coordinate values in the input sets are handled by delaying the addition of points to L_t until they are removed from L_t^* as a result of the insertion of a new point (lines 4–5). This guarantees that output points generated in one iteration of the outer loop which become dominated in a subsequent iteration due to the processing of a second input point with the same z coordinate are not actually added to the output. Lines 4–5 do not apply to the updating of the X_j^* data structures, which is otherwise identical.

Complexity. A complete run of the algorithm involves m searches and (up to) m insertions into the X_j^* data structures, plus up to m search-removal pairs, to maintain the input data structures. Since the total number of input points is m, the cost of these operations is bounded by $O(\log m)$, and the complexity due to them is $O(m \log m)$.

Maintaining the output data structures, L_t^*, requires $O(n^2 m \log(n^2 m)) = O(n^2 m \log m)$ time,[3] since there are $O(n^2 m)$ output points. In addition, $O(m)$ searches in the X_j^* data structures and $O(n^2 m)$ searches in the L_t^* data structures are performed in the body of the algorithm, including $O(nm)$ searches in L_t^* that may not lead to the generation of new output points. As this extra work is also done in $O(n^2 m \log m)$ time, the time complexity of Algorithm 4 is $O(n^2 m \log m)$, and the algorithm is asymptotically optimal as long as the number of input sets, n, is assumed to be constant (compare with Proposition 5). When $m \in O(n)$, the time complexity of the algorithm is only a logarithmic factor worse than the cardinality upper bound derived in Section 4, Proposition 3.

6 Concluding Remarks

In this work, the EAF computation problem has been formalised as the problem of computing a finite description of all of its superlevel sets (or the corresponding summary attainment surfaces) from experimental data. After proving that each component of the solution of this problem consists of the set of minima of a suitable auxiliary set constructed based on the input data, the size of the output sets which describe the EAF was shown to grow linearly with the number of input points only when $d = 2, 3$. Finally, efficient algorithms for the EAF in two and three dimensions were developed based on existing dimension-sweep algorithms for minima. The algorithm for $d = 3$ is asymptotically optimal when the number of input sets is fixed, whereas the algorithm for $d = 2$ is asymptotically optimal in the general case.

Extending the algorithms proposed here to more than three dimensions is the subject of further work, although quadratic complexity with respect to the number of input points will be unavoidable in the worst case. This is a direct consequence of the lower bound stated in Proposition 5, but it may still be possible to seek improved performance in non-worst-case scenarios, e.g., by developing output-sensitive EAF algorithms.

A C-language implementation of Algorithms 3 and 4 is available from the authors on `http://eden.dei.uc.pt/~cmfonsec/software.html`. As an example of practical runtime performance, computing the EAF of a set of 50 spherical, three-objective fronts with 240 points each took only 6 seconds on a single core of an AMD Opteron 2216 HE 2.4GHz processor. It is expected that these results will contribute to a more widespread use of the empirical attainment function as a performance assessment tool in Evolutionary Multiobjective Optimisation.

Acknowledgements. Andreia P. Guerreiro acknowledges financial support from CEG-IST through an Integration into Research Grant sponsored by the Portuguese Foundation for Science and Technology. Manuel López-Ibáñez acknowledges support from the FRFC project "*Méthodes de recherche hybrides pour la résolution de problèmes complexes*".

[3] Note that $n \leq m$.

References

1. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A platform and programming language independent interface for search algorithms. In: Fonseca, C.M., et al. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
2. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Voigt, H.M., et al. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 584–593. Springer, Heidelberg (1996)
3. Fonseca, C.M., Grunert da Fonseca, V., Paquete, L.: Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello, C.C., et al. (eds.) EMO 2005. LNCS, vol. 3410, pp. 250–264. Springer, Heidelberg (2005)
4. Grunert da Fonseca, V., Fonseca, C.M.: The attainment-function approach to stochastic multiobjective optimizer assessment and comparison. In: Bartz-Beielstein, T., et al. (eds.) Experimental Methods for the Analysis of Optimization Algorithms. pp. 103–130. Springer, Berlin (2010)
5. Grunert da Fonseca, V., Fonseca, C.M., Hall, A.O.: Inferential performance assessment of stochastic optimisers and the attainment function. In: Zitzler, E., et al. (eds.) EMO 2001. LNCS, vol. 1993, pp. 213–225. Springer, Heidelberg (2001)
6. Knowles, J.D.: A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, pp. 552–557 (2005)
7. Kung, H.T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. Journal of the ACM 22(4), 469–476 (1975)
8. López-Ibáñez, M., Paquete, L., Stützle, T.: Exploratory analysis of stochastic local search algorithms in biobjective optimization. In: Bartz-Beielstein, T., et al. (eds.) Experimental Methods for the Analysis of Optimization Algorithms, pp. 209–222. Springer, Berlin (2010)
9. Preparata, F.P., Shamos, M.I.: Computational Geometry. An Introduction, 2nd edn. Springer, Berlin (1988)
10. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms – A comparative case study. In: Eiben, A.E., et al. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)

Computing Hypervolume Contributions in Low Dimensions: Asymptotically Optimal Algorithm and Complexity Results

Michael T.M. Emmerich[1,2] and Carlos M. Fonseca[1,2,3]

[1] DEEI, Faculty of Science and Technology, University of Algarve
Campus de Gambelas, 8005-139 FARO, Portugal
mtemmerich@ualg.pt

[2] CEG-IST, Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal

[3] Department of Informatics Engineering, University of Coimbra,
Pólo II, 3030-290 Coimbra, Portugal
cmfonsec@dei.uc.pt

Abstract. Given a finite set $\mathrm{Y} \subset \mathbb{R}^d$ of n mutually non-dominated vectors in $d \geq 2$ dimensions, the hypervolume contribution of a point $\mathbf{y} \in \mathrm{Y}$ is the difference between the hypervolume indicator of Y and the hypervolume indicator of $\mathrm{Y} \setminus \{\mathbf{y}\}$. In multi-objective metaheuristics, hypervolume contributions are computed in several selection and bounded-size archiving procedures.

This paper presents new results on the (time) complexity of computing all hypervolume contributions. It is proved that for $d = 2, 3$ the problem has time complexity $\Theta(n \log n)$, and, for $d > 3$, the time complexity is bounded below by $\Omega(n \log n)$. Moreover, complexity bounds are derived for computing a single hypervolume contribution.

A dimension sweep algorithm with time complexity $\mathcal{O}(n \log n)$ and space complexity $\mathcal{O}(n)$ is proposed for computing all hypervolume contributions in three dimensions. It improves the complexity of the best known algorithm for $d = 3$ by a factor of $\sqrt{n}$. Theoretical results are complemented by performance tests on randomly generated test-problems.

Keywords: multiobjective selection, complexity, hypervolume indicator.

1 Introduction

The *hypervolume indicator* (or S-metric, Lebesgue measure) was introduced by Zitzler and Thiele [26] to measure the quality of Pareto front approximations. Given a finite set Y of mutually non-dominated vectors in $\mathbb{R}^d$, the hypervolume indicator measures the volume (Lebesgue measure) of the subspace simultaneously dominated by Y and bounded below (in case of maximization) by a reference point.

Besides being frequently used as a performance metric, the hypervolume indicator is also used in guiding selection in indicator-based metaheuristics

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 121–135, 2011.

[10,11,13,14,19,25] and archivers [16]. In this context, the problem of computing hypervolume contributions and/or the minimal hypervolume contributor of a set of points often arises [10,13,14,16,20,25]. The hypervolume contribution of a point $\mathbf{y} \in \mathrm{Y}$ is defined as the difference between the hypervolume indicator of Y and the hypervolume indicator of $\mathrm{Y} \setminus \{\mathbf{y}\}$. The problem of finding the minimal hypervolume contributor is #P-hard in the number of dimensions d [8].

Many applications of multiobjective optimization involve a small number of objectives, say 2–4. In these cases, polynomial time algorithms for computing hypervolume contributions are known (e.g. [9,20]), but the extent to which they are efficient is so far unknown. When objective functions can be computed fast, hypervolume computations can limit the performance of multiobjective optimization algorithms also in low dimensions.

Hence, this paper focuses on computing hypervolume contributions in low dimensions and analyzes complexity in the cardinality, n, of the input set, Y. One aim is to derive sharper complexity bounds for the computation of all (or single) hypervolume contributions. Moreover, it is shown that the dimension sweep paradigm, that has yielded asymptotically optimal algorithms for computing the hypervolume indicator [4] and maximal set [18] in low dimensions, can also allow hypervolume contributions to be computed efficiently.

The paper is organized as follows: In Section 2, several problems related to hypervolume contributions are introduced. A summary of known results and related work on these problems is given. In Section 3, lower bounds on the complexity of computing hypervolume contributions are established. In Section 4, an algorithm that efficiently computes the hypervolume contributions given a set in three dimensions is introduced and analyzed. In Section 5, performance tests on randomly generated test data are conducted. Finally, in Section 6, contributions of the paper are summarized and suggestions for future research are given.

2 Preliminaries and Related Work

The following conventions will be used throughout the paper. Complexity, unless stated otherwise, refers to *time complexity in the worst case*. The algebraic decision tree model of computation is considered [17]. Asymptotic lower, upper, and sharp bounds are denoted by $\Omega()$, $\mathcal{O}()$, and $\Theta()$, respectively. Vectors are represented in bold font. Sets are denoted by roman capitals, e.g. X or Y, and problems and algorithm names are typeset in small capitals, e.g. allContributions. When applied to vectors, operators $\leq$, $\geq$ and $=$ indicate componentwise comparison. Throughout this paper, maximization is the goal.

The following concepts will be relevant in the problem definition:

Definition 1. *(dominance) A vector* $\mathbf{y} \in \mathbb{R}^d$ dominates $\mathbf{y}' \in \mathbb{R}^d$*, iff* $\mathbf{y} \geq \mathbf{y}'$ *and* $\mathbf{y} \neq \mathbf{y}'$*. In symbols:* $\mathbf{y} \succ \mathbf{y}'$*.*

Dominance is often defined with minimization of vectors as the goal. In this case $\geq$ must be replaced by $\leq$ in the above definition.

Definition 2. *(approximation set) A d-dimensional approximation set [27] is a finite set* $\mathrm{Y} \subset \mathbb{R}^d$ *such that* $\forall \mathbf{y}, \mathbf{y}' \in \mathrm{Y} : \mathbf{y} \geq \mathbf{y}' \Rightarrow \mathbf{y} = \mathbf{y}'$*. The set of all d-dimensional approximation sets is denoted by* $\mathbb{A}^d$*.*

Definition 3. *(hypervolume indicator ($\mathcal{S}$)) Given a set* $\mathrm{Y} \subset \mathbb{R}^d$ *and a reference point* $\mathbf{y}^r$ *that satisfies* $\forall \mathbf{y} \in \mathrm{Y} : \mathbf{y} \geq \mathbf{y}^r$*, the hypervolume indicator of* Y *with respect to reference point* $\mathbf{y}^r$ *is defined as [26,24]:*

$$\mathcal{S}(\mathrm{Y}) = \mathrm{Vol}\left(\bigcup_{\mathbf{y} \in \mathrm{Y}} [\mathbf{y}^r, \mathbf{y}]\right). \tag{1}$$

Here $\mathrm{Vol}()$ *denotes the Lebesgue measure in d dimensions and* $[\mathbf{l}, \mathbf{u}] \subset \mathbb{R}^d$ *represents a closed axis-parallel box that is bounded below by* $\mathbf{l} \in \mathbb{R}^d$ *and bounded above by* $\mathbf{u} \in \mathbb{R}^d$*.*

Whenever the reference point is not explicitly mentioned, by convention, $\mathbf{y}^r = \mathbf{0}$ will be assumed.

Definition 4. *(hypervolume contribution of a point) Given an approximation set* $\mathrm{Y} \in \mathbb{A}^d$*, the hypervolume contribution of a point* $\mathbf{y} \in \mathrm{Y}$ *is defined as*

$$\Delta\mathcal{S}(\mathbf{y}, \mathrm{Y}) = \mathcal{S}(\mathrm{Y}) - \mathcal{S}(\mathrm{Y} \setminus \{\mathbf{y}\})$$

This paper is mainly concerned with the following problem:

*Problem 1 (*AllContributions*).* Given $\mathrm{Y} \in \mathbb{A}^d$ as an input, compute the hypervolume contributions of all points $\mathbf{y} \in \mathrm{Y}$.

In addition new results on closely related problems, that can all be reduced in at most linear time to AllContributions, will be derived:

*Problem 2 (*OneContribution*).* Given $\mathrm{Y} \in \mathbb{A}^d$ and $\mathbf{y} \in \mathrm{Y}$ as an input, compute the hypervolume contribution $\Delta\mathcal{S}(\mathbf{y}, \mathrm{Y})$.

*Problem 3 (*MinimalContribution*).* Given $\mathrm{Y} \in \mathbb{A}^d$ as an input, compute the minimal hypervolume contribution, i.e., $\min_{\mathbf{y} \in \mathrm{Y}} \Delta\mathcal{S}(\mathbf{y}, \mathrm{Y})$.

*Problem 4 (*MinimalContributor*).* Given $\mathrm{Y} \in \mathbb{A}^d$ as an input, find a point with minimal hypervolume contribution, i.e., $\mathbf{y}^* \in \arg\min_{\mathbf{y} \in \mathrm{Y}} \Delta\mathcal{S}(\mathbf{y}, \mathrm{Y})$.

A straightforward algorithm to solve AllContributions consists of enumerating all subsets of size $n - 1$, computing their hypervolumes, and subtracting each of these in turn from the hypervolume of the total set. Bringmann and Friedrichs [7] show that computing the hypervolume is #P-hard in the number of dimensions d. To compute the hypervolume, for $d = 3$, the dimension sweep algorithm by Fonseca et al.[12] is asymptotically optimal (Beume et al.[4]). For general d, computing the hypervolume indicator can be considered a special case of computing the measure of a union of rectangles [3]. The best known algorithm has a complexity $\mathcal{O}(n^{d/2} \log n)$ (cf. [21]) and has been simplified for the special case of computing hypervolume by Beume [3], though the complexity stays the same. A dimension sweep algorithm with complexity $\mathcal{O}(n^{d-2} \log n)$ is described by Fonseca et al. [12]. It has lower complexity for $d = 3$ and the same complexity for $d = 4$. Algorithms with a higher worst case complexity proposed in While et al. [23] have proved to be competitive or faster on problems of moderate size and dimensions.

A specialized algorithm for computing hypervolume contributions that guarantees a better worst-case performance has been proposed by Bringmann and Friedrichs [9] based on Overmars and Yap's algorithm, [21]. Its time complexity is $\mathcal{O}(n^{d/2} \log n)$ for $d > 2$. Note, that their algorithm also solves the more general problem of finding the hypervolume contributions of a finite set $\mathrm{Y} \in \mathbb{R}^d$, i.e. it is not required that points in the set are mutually non-dominated. For $d = 2$, the problem reduces simply to sorting a set of n points ($\mathcal{O}(n \log n)$) (e.g. [16]). Another algorithm for computing *incremental hypervolumes* has been proposed by Bradstreet et al. [6], who also describe an algorithm for updating all contributions in [5], the worst case complexity of which is $\mathcal{O}(n^{d-1})$. Empirical studies show a high performance of these schemes in case of many objectives.

For high dimensions, fast heuristics and approximation algorithms have been proposed. For instance, Bader and Zitzler [2] propose Monte Carlo algorithms that work also for many objectives. Fast approximation algorithms with accuracy guarantees are proposed by Bringmann and Friedrichs [8]. Approximate integration based on scalarization is suggested in [15].

Despite this progress, sharp complexity bounds for AllContributions, OneContribution, MinimalContribution, and MinimalContributor have remained unavailable to date.

3 Complexity Bounds

This section will start with a theorem on a lower bound on AllContributions and proceed with the discussion of lower bounds on computing single contributions. To obtain a lower bound on the complexity of AllContributions a reduction to UniformGap, a well-known problem from computational geometry, is used.

*Problem 5 (*UniformGap). The problem of deciding whether a set of n points on the real line is equidistant is called UniformGap.

Lemma 1. *(Preparata and Shamos [22], p. 260) The complexity of* UniformGap *is $\Omega(n \log n)$ in the algebraic decision tree model of computation.*

Now, the theorem on the complexity of AllContributions reads:

Theorem 1. *Any algorithm that solves* AllContributions *in $d > 1$ dimensions requires $\Omega(n \log n)$ time in the algebraic decision tree model of computation.*

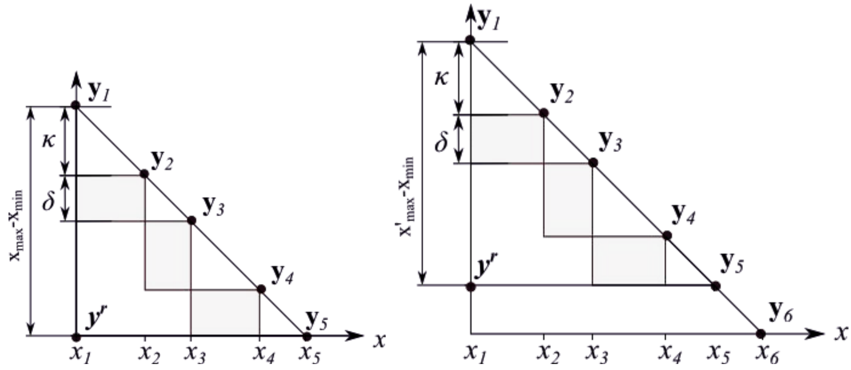


Fig. 1. Problem transformation. Odd cardinality (left) and even cardinality (right)

Proof of theorem 1: It will be shown that UniformGap reduces in linear time to AllContributions. (For the general idea of a reduction proof see, e.g., [4].) Given a set X of n values x_i on the real line as input to UniformGap, each coordinate is augmented with a second coordinate, which yields $\mathrm{Y} = \{(x_i, x_{max} - x_i) : i \in \{1, \dots, n\}$, where $x_{max} = \max\{x_i : i = 1, \dots, n\}$ (see Fig. 1).

Case 1 (n is odd): Compute all hypervolume contributions of points using $\mathbf{y}^r = (x_{min}, x_{min})$, $x_{min} = \min(X)$ as reference point, see Fig. 1. Next, all hypervolume contributions are computed and compared to $((x_{max} - x_{min})/(n-1))^2$. If they are all equal to this value the answer to UniformGap is positive, otherwise it is negative. This algorithm is correct for odd n, since all hypervolume contributions of inner points are equal to $((x_{max} - x_{min})/(n-1))^2$ if and only if gaps are uniform. This will be shown next: Consider a line segment from $\mathbf{y}_1 = (x_{min}, x_{max})$ to $\mathbf{y}_n = (x_{max}, x_{min})$. As all intermediate points in Y lie on the line segment, the hypervolume contribution equality condition enforces a certain alternating pattern of rectangles bounding the contributions (see Fig. 1). The rectangles are congruent, but rotate by 90° in each step. The side-length of the rectangles is δ for the shorter side and κ for the longer side. Because the number of points on the line segment is odd

$$x_{max} - x_{min} = \frac{n-1}{2}(\delta + \kappa) \text{ and thus } \kappa = \frac{2(x_{max} - x_{min})}{(n-1)} - \delta. \tag{2}$$

The contribution of each inner point x is given by $c(x) = \delta\kappa$. To maximize c over all δ and κ eliminate κ in $c(x) = \delta\kappa$, which leads to the problem:

$$\text{maximize } c(\delta) = \delta\left(\frac{2(x_{max} - x_{min})}{n-1} - \delta\right) \tag{3}$$

The equation describes a concave parabola, the unique maximizer δ^* of which is the point with zero derivative. Solving $2\frac{(x_{max} - x_{min})}{n-1} - 2\delta^* = 0$ yields

$$\delta^* = \frac{x_{max} - x_{min}}{n-1} \tag{4}$$

Inserting δ^* in Equation 3 yields $\kappa^* = \delta^*$. This solution is the only solution with (maximal) hypervolume contribution $c = (\delta^*)^2 = ((x_{max} - x_{min})/(n-1))^2$ for all points $\mathrm{Y} \setminus \{\mathbf{y}_1, \mathbf{y}_n\}$.

Case 2 (n is even): Let $\mathrm{X}' = \mathrm{X} \setminus \{\max(\mathrm{X})\}$. Let $x'_{max} = \max(\mathrm{X}')$ and $x'_{min} = x_{min}$. Given that the uniform gap condition is fulfilled for X' (odd size) and $x_{max} - x'_{max} = (x'_{max} - x'_{min})/(n-2)$ the answer is positive, otherwise not. The complexity of the extra steps in this algorithm is $\mathcal{O}(N)$, excepting perhaps the time for solving AllContributions. Thus, if AllContributions could be solved faster than $\mathcal{O}(n \log n)$, then also UniformGap could be solved faster than $\mathcal{O}(n \log n)$, which would contradict lemma 1.

The result generalizes to $d > 2$ as otherwise the problem AllContributions in two dimensions could be solved by a linear time reduction to AllContributions in higher dimensions. It would suffice to set all $d-2$ additional coordinates of $\mathbf{y}$ to 1 and the reference point to $\mathbf{0}$ and solve the problem. □

A similar result can be obtained for computing single contributions:

Theorem 2. *The time complexity of* OneContribution *is $\Theta(n)$ for $d = 2$, $\Theta(n \log n)$ for $d = 3$, and for $d > 3$ it is bounded below by $\Omega(n \log n)$.*

Proof: Case $d = 2$: All points need to be considered, yielding $\Omega(n)$ as lower bound; computing a contribution requires $\mathcal{O}(n)$ time (nearest neighbor search). Case $d = 3$: The problem of computing the hypervolume of a set in two dimensions (Hypervolume2d) can be reduced in linear time to OneContribution in three dimensions. The complexity of Hypervolume2D is bounded below by $\Omega(n \log n)$ (cf. [4]). To compute the solution of Hypervolume2D for an $\mathrm{Y} \in \mathbb{A}^2$ using OneContribution in 3-D, represent Y in three dimensions by augmenting all points Y with a z-coordinate of 2, resulting in $\mathrm{Z} \in \mathbb{A}^3$. Create a point $(x_{max}, y_{max}, 1)$ in $\mathcal{O}(n)$ time with x_{max} being the maximum x-coordinate and y_{max} the maximum y-coordinate in Y. The solution of Hypervolume2D is $x_{max}y_{max} - \Delta\mathcal{S}((x_{max}, y_{max}, 1), (x_{max}, y_{max}, 1) \cup \mathrm{Z})$. The same principle can be used for proving a lower bound of $\Omega(n \log n)$ for $n > 3$. The upper bound of $\mathcal{O}(n \log n)$ for $n = 3$ is due to the fact that the computation of a single contribution of a point $\mathbf{y}$ in Y reduces to computing the difference $\mathcal{S}(\mathrm{Y}) - \mathcal{S}(\mathrm{Y} \setminus \{\mathbf{y}\})$, and the computation of each hypervolume takes $\mathcal{O}(n \log n)$ (see [12]). □

4 Dimension Sweep Algorithm

A dimension sweep algorithm that computes the hypervolume contributions for a set $\mathrm{Y} \in \mathbb{A}^3$ in $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space is introduced and discussed. The dimension sweep paradigm seems to be promising for constructing the algorithm, as it has already yielded asymptotically optimal algorithms for the maximal set problem [18] and the computation of the total hypervolume in 3-D [12].

The algorithm processes points in the order of descending z-coordinates. For the i-th such level, the z-coordinate of the i-th point defines the height of a sweeping plane that stretches out in x- and y-direction (see figure 2).

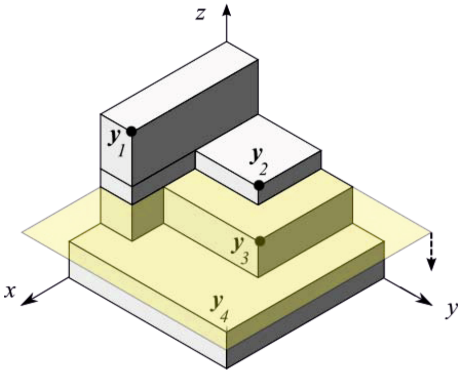


Fig. 2. Sweeping plane

The construction of the algorithm is based on the observation that the region above the sweeping plane that is exclusively dominated by a point when considering only the subset of points above the sweeping plane, will be also exclusively dominated – by the same point – when the entire set of points is considered. Hence hypervolume contributions can be accumulated level by level. This can be done by partitioning the exclusively dominated hypervolume into axis-parallel boxes. At each level, it is made sure that each exclusively dominated region above the sweeping plane is covered by exactly one such box. As the sweeping plane reaches lower levels new boxes may be *created*, existing boxes may grow (in their z-coordinate), and boxes may be *completed*, i.e., they will no longer grow. Each box is associated with the point that exclusively dominates the hypervolume it encloses. After the sweeping plane has reached the level of the reference point, the exclusive hypervolume contribution of a point is the sum of the volumes of the completed boxes that have been associated with this point.

To achieve efficiency, the algorithm maintains lists of boxes in such a way that each box that needs to be updated can be efficiently accessed. This can be achieved by maintaining boxes in doubly linked lists associated with the points. Only *active* boxes are maintained, i.e. boxes that may still grow. Boxes that are completed are removed from these lists, and their hypervolume is added to the point to which they belong. Moreover, efficient access to points, the box lists of which need to be updated, is achieved by maintaining a tree that comprises a subset of points that have already been visited [18]. Accordingly, the following non-elementary *data-structures* are used in the algorithm:

– A height balanced search tree (e.g. AVL tree [1]), named T, the nodes of which refer to the non-dominated subset of input points that have been processed so far by the algorithm. It is sorted in ascending order of the x-coordinate and features insertion of a point (T.insert), deletion of a point (T.delete) (both in time $\mathcal{O}(\log n)$), and range selection in $\mathcal{O}(\log(n) + k)$, where n is the number of points and k the number of qualified items.

- An axis-parallel box **b** is defined by its lower corner (**b**.lx, **b**.ly, **b**.lz), and its upper corner (**b**.ux, **b**.uy, **b**.uz). *Active* boxes have an undefined lower bound **b**.lz=`NaN` whereas all other coordinates are set to their final value after a box has been created.
- An array of doubly linked lists of active boxes: Each point is associated with one box list. List are sorted in ascending x-coordinate. A box list, say L, has these methods: L.push_front(**b**) adds a box item to the list's head, and L.push_back(**b**) to its tail. Moreover, **b**=L.pop_front() retrieves a box from the head of a list and **b**=L.pop_back() from the tail, removing the corresponding box from the list.

A detailed outline of the algorithm in pseudocode is given in algorithm 1. The array of points is sorted by the z-coordinate. Sorting can be achieved in $\mathcal{O}(n \log n)$ and therefore, due to theorem 1, it does not impose significant additional costs onto the algorithm.

Algorithm 1 can be subdivided in an initialization phase and the main loop. *Initialization:* The algorithm intializes an empty AVL tree (T) and inserts the highest point **p**[1] into T. Also, two auxiliary points $(\infty, 0, \infty)$ and $(0, \infty, \infty)$ that bound the region from the left and from the right are inserted. They have either an x or a y coordinate of 0, respectively, such that they never dominate points with positive hypervolume contribution. Here, the value of 0 stems from the reference point $\mathbf{c} = \mathbf{0}$. Their z-coordinate is infinite, such that they are never dominated themselves by input points and remain in the tree. Throughout the algorithm they serve to make sure that every point in the xy-plane has a well defined neighbor in the x- and in the y-coordinate.

Additionally, an auxiliary point $\mathbf{p}[n+1]$=(∞, ∞, 0) is created. It receives the index $n + 1$. It will be processed in the last iteration of the main loop. Again, the value of 0 stems from the reference point. Its projection onto the xy-plane is (∞, ∞) and dominates all projections of input points to the xy-plane. This point will force the completion of boxes that are still open in the last iteration. The array L of box lists is then initialized with empty lists. A point is inserted into the box list of point **p**[1] with (**p**[1].x, **p**[1].y, **p**[1].z) and the x-and y-coordinate of the reference point as a lower corner. The lower limit of the box in the z-coordinate is determined later in the algorithm. This box delimits the region that is dominated exclusively by the first point above the second level of the sweeping plane. Finally, the array c[1], ..., c[n] that is used to accumulate hypervolume contributions of completed boxes is initialized to zero.

The *main loop* of the algorithm processes in ascending order of the z-coordinate all points **p**[2], ... $\mathbf{p}[n+1]$. It maintains the following invariant properties:

- At the end of each iteration, the volume of all active boxes in the box lists, if they were completed at the current z-level, plus the accumulated volumes of previously completed boxes, is the total exclusively dominated hypervolume above the sweeping plane at level i. This property is essential to guarantee the correctness of the algorithm.
- The box lists of each point contains active boxes sorted by ascending x coordinates. This property allows efficient updates of box lists.

Algorithm 1. Algorithm HYCON3D

```
    Input: (p[1], ..., p[$n$]): mutually non-dominated $\mathbb{R}^3$-points sorted by $z$-coordinate
    in descending order
    (1) p[n+1]=$(\infty, \infty, 0)$;
    (2) Initialize AVL tree T for 3-D points
        Insert p[1], $(\infty, 0, \infty)$, $(0, \infty, \infty)$ into T;
    (3) Initialize doubly linked lists L[1]=empty(); ... L[n+1]=empty();
        b= ((0,0,NaN), (p[1].x, p[1].y, p[1].z)); L[1].push_front(b);
    (4) Initialize hypervolume contributions $c[1] = 0$; ...; $c[n] = 0$
    for i= 2 to n+1 do {Main Loop}
    (a) Retrieve the following information from tree T:
          r: index of the successor of p[i] in x-coordinate (right neighbor)
          t: index of the successor of p[i] in y-coordinate (left neighbor)
          d[1], ..., d[s]: indices of points dominated by p[i] in xy-plane,
          sorted ascending in $x$-coordinate (region B, Figure 3).
    (b) while not L[r].empty()    {Process right neighbor, region R}
            b= L[r].pop_front()
            if (b.ux $\leq$ p[i].x)
                b.lz=p[i].z; c[r]=c[r]+VOL(b);
            else if (b.lx $<$ p[i].x)
                b.lz=p[i].z; c[r]=c[r]+VOL(b);
                b.lx=p[i].x; b.uz=p[i].z; b.lz=NaN; {Add box $b^r$ in region R}
                L[r].push_front(b); break;
            else L[r].push_front(b); break
    (c) xleft = p[t].x    {Process dominated points, region M}
        for j=1 to s
            jdom=d[s]; d=p[jdom];
            while (not L[jdom].empty())
                b=L[jdom].pop_front();
                b.lz=p[i].z; c[jdom]=c[jdom]+VOL(b);
            b=[(xleft, d.y, NaN),(d.x, p[i].y, p[i].z)];
            L[i].push_back(b);
            xleft = b.ux;
            remove p[jdom] from AVL tree
        b=[(xleft, p[r].y, NaN), (p[i].x, p[i].y, p[i].z)]; {Add box $b^+$ in region R}
        L[i].push_back(b);
    (d) xleft=p[t].x;    {Process left neighbor, region L}
        while not L[t].empty()
            b=L[t].pop_back();
            if (b.ly $<$ p[i].y)
                b.lz = p[i].z; c[t]=c[t]+VOL(b);
                xleft = b.lx
            else L[t].push_back(b); break;
        if (xleft $<$ p[t].x)
            b=[(xleft, p[i].y, NaN),(p[t].x, p[t].y, p[i].z)]; {Add box $b^l$ in region R}
            L[t].push_back(b);
    (e) Insert p[i] in AVL tree T;
(5) output c[1], ..., c[n] {Exclusive contributions}
```

– To find relevant information for the update of box lists, in the i-th iteration, T stores points that are non-dominated among $\mathbf{p}[1]$, ..., $\mathbf{p}[i-1]$ in the xy-plane. These are the only relevant points for determining increments of hypervolume contributions from level i onwards. T is organized as a balanced search tree sorted by x-coordinate.

After the introduction of a new point $\mathbf{p}[i]$ some boxes simply grow in z-direction and no access by the algorithm is required for them, whereas others need to be created or completed and require access. From Figure 3, it becomes clear that for the creation and completion of boxes only box lists of particular points are of interest. These are $\mathbf{p}[r]$, the upper neighbor in T of $\mathbf{p}[i]$ in the x-direction, $\mathbf{p}[t]$, the upper neighbor in T of $\mathbf{p}[i]$ in the y-direction, and the sequence of all dominated points $\mathbf{p}[\mathrm{d}[1]]$, ..., $\mathbf{p}[\mathrm{d}[s]]$, being in ascending order of the x-coordinate. The algorithm determines the indices of these points using the AVL tree and making use of the fact that a sorting in the x-coordinate implies a reverse order in the y-coordinate (points in the AVL tree are mutually non-dominated in the xy-plane.)

Firstly, the right hand side of $\mathbf{p}[i]$ is considered (region R in Figure 3). The point $\mathbf{p}[r]$ may dominate regions exclusively until level i that from this level onwards are no longer dominated exclusively by $\mathbf{p}[r]$. They can be obtained step-by-step by traversing the box list $\mathrm{L}[r]$ in the direction of ascending x-coordinates (from the head of the list). Each dominated box is completed and the yet undefined z-coordinate is set to $\mathbf{p}[i].z$. The volume of the box is added to the hypervolume contribution $\mathrm{c}[r]$ of $\mathbf{p}[r]$. The algorithm stops removing boxes from the list after the lower bound of a box retrieved from L[r] exceeds $\mathbf{p}[i].x$. The right part of the last box removed may still be exclusively dominated by $\mathbf{p}[r]$. Hence, a new box, (b^r, in Figure 3), may be inserted and attached to the front of the list L[r].

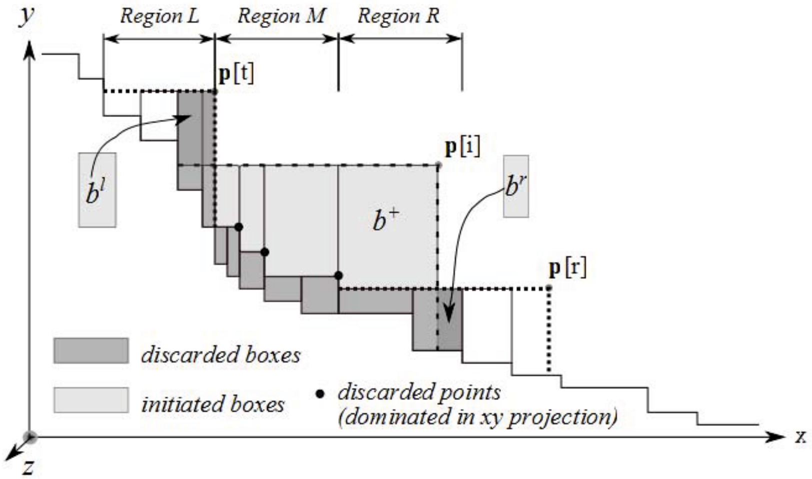


Fig. 3. Single level of the sweeping plain

Region M (see Figure 2) comprises the points dominated by **p**[i] in the xy-plane, namely **p**[d[1]], ..., **p**[d[s]]. They are processed in ascending order of x. For each such point, a new box is created and pushed to the back of the list of **p**[i]. Additionally, a box in region R is created and pushed to the back of the same list with **p**[i] as an upper corner (b^{+} in Figure 3), if **p**[d].x < **p**[i].x. Its lower y-coordinate is **p**[r].y. After adding these boxes, the area that is, from now on, exclusively dominated by **p**[i] is partitioned into boxes. Moreover, all boxes associated with dominated points (projections) are completed and discarded from the box lists, as they are from now on also dominated by **p**[i] in the xy-plane. The dominated points themselves are discarded from the tree T for the same reason.

Boxes to the left of **p**[t] (in region L in Figure 3) are only dominated by **p**[t] up to the i-th level, and from level i onwards, additionally dominated partly by **p**[i]. The boxes in this region need to be updated in their lower y bound. Above **p**[i].y they will still be exclusively dominated by **p**[t]. The update is achieved by completing all boxes with lower y-bound smaller than **p**[i].y. The corresponding regions exclusively dominated by **p**[t] can be merged into one single box (b^{l} in Figure 3). Therefore only one new box is added to **p**[t]. This box has **p**[i].y as its lower bound in the y-direction, and **p**[t].y as its upper bound.

In the final iteration of the algorithm, $i = n + 1$ and all previously processed points are dominated by the auxiliary point **p**[$n+1$]. Volumes are updated for all points that still have boxes in their lists. These boxes are lower bounded in the z-coordinate by the reference point. After this operation has finished, the array c[1], ..., c[n] contains the exclusive contributions of the points **p**[1], ..., **p**[n].

The runtime of algorithm HyCon3D is estimated by the following theorem:

Theorem 3. *Algorithm* HyCon3D *has a time complexity of* $\mathcal{O}(n \log n)$.

Proof: The time for the initialization phase, including sorting the input, is bounded by $\mathcal{O}(n \log n)$. Time critical operations within the main loop are (1) updating and retrieving points from the AVL tree, and (2) creation and completion of boxes.

The algorithm needs to retrieve all dominated points, and the neighbor point indices r and t from the AVL tree. To identify the neighbor to the right takes time $\mathcal{O}(\log n)$. As the xy-projection is mutually non-dominated the points are sorted also by their y coordinate, and the tree can then be traversed in constant time per additional dominated point. Each point is inserted and discarded only once. The total cost of these operations amortizes to $\mathcal{O}(n \log n)$. Excepting the two boundary points **p**[r] and **p**[t], each point that is retrieved from the tree is discarded. Hence, the total cost of tree operations also amortizes to $\mathcal{O}(n \log n)$.

Furthermore, the total number of boxes that are created and completed can be bounded. Each box is created and completed only once. Four different events can cause the creation of a box. Each point **p** that is processed in the algorithm causes the creation of (at most) two boxes: One box might be created when the **p** is inserted, of which **p** is the upper corner. Moreover a box is created when **p** gets discarded. It is associated with the point that at that time dominates **p** in the xy-plane. In total, at most $2n$ boxes are created this way.

At most two additional boxes may be created per level i, one box for $\mathbf{p}[\mathrm{t}]$ and one box for $\mathbf{p}[\mathrm{r}]$. Accumulating over all n levels, at most $2n$ such boundary boxes are created. Hence, at most $4n$ boxes are created in total.

For all box creations and completions, any relevant point and any box in that point's list of boxes can be located in constant time, because boxes that need to be inserted and deleted are always at the head or at the tail of a doubly linked box list. This results in an overall cost for box list updates of $\mathcal{O}(n)$, and as volumes are only updated when a box is deleted, this is also the complexity of all hypervolume contribution updates. As no other time critical operations are executed, the complexity is $\mathcal{O}(n \log n)$. □

Theorem 3 implies sharper bounds on the complexity of related problems:

Theorem 4. *The following statements hold for $d = 3$ and an input set $\mathrm{Y} \in \mathbb{A}^d$ of size n:*

1. AllContributions *has time complexity $\Theta(n \log n)$.*
2. *The time complexity of* MinimalContribution *is bounded by $\mathcal{O}(n \log n)$.*
3. *The time complexity* MinimalContributor *is bounded by $\mathcal{O}(n \log n)$.*

Proof: (1) Theorem 3 establishes an upper bound for AllContributions of $\mathcal{O}(n \log n)$. This matches the lower bound of $\Omega(n \log n)$ (see theorem 1). (2)+(3): Finding a minimal contribution and contributor can be accomplished in a single scan (linear time) of the results of AllContributions. □

5 Numerical Experiments

Experiments are conducted in order to find out the CPU-time needed to compute all hypervolume contributions for input sets of different size $n \in [100, 1000]$. For each value of n, the time needed to process $m = 100$ randomly generated Pareto fronts is measured. Performance is studied on three test problems:

*Problem 6 (*convexSpherical). Find all contributions of $\mathrm{Y} \in \mathbb{A}^d$, where $\mathbf{y}$ are generated independently, and $y_i = 10|v_i|/\|\mathbf{v}\|$, $v_i \sim \mathrm{Normal}(0,1), i = 1, \ldots, d$. (cf. Figure 4, upper left)

*Problem 7 (*concaveSpherical). Find all contributions of $\mathrm{Y} \in \mathbb{A}^d$, where $\mathbf{y} \in \mathrm{Y}$ are generated independently, and $y_i = 10 - 10|v_i|/\|\mathbf{v}\|$, $v_i \sim \mathrm{Normal}(0,1), i = 1, \ldots, d$. (cf. Figure 4, upper middle)

*Problem 8 (*cliff3D). Find all contributions for $\mathrm{Y} \subset \mathbb{A}^d$, where $\mathbf{y} \in \mathrm{Y}$ are generated independently, and $y_i = 10|v_i|/\|\mathbf{v}\|$, $v_i \sim \mathrm{Normal}(0,1)$, $i = 1, 2$, $y_3 \sim \mathrm{Uniform}(0, 10)$. (cf. Figure 4, upper right)

convexSpherical and concaveSpherical consider uniformly distributed points on a convex and, respectively, concave spherical surface. cliff3D considers the third coordinate according to a uniform distribution. It is constructed in a way that, during the dimension sweep, all points remain in the tree until the

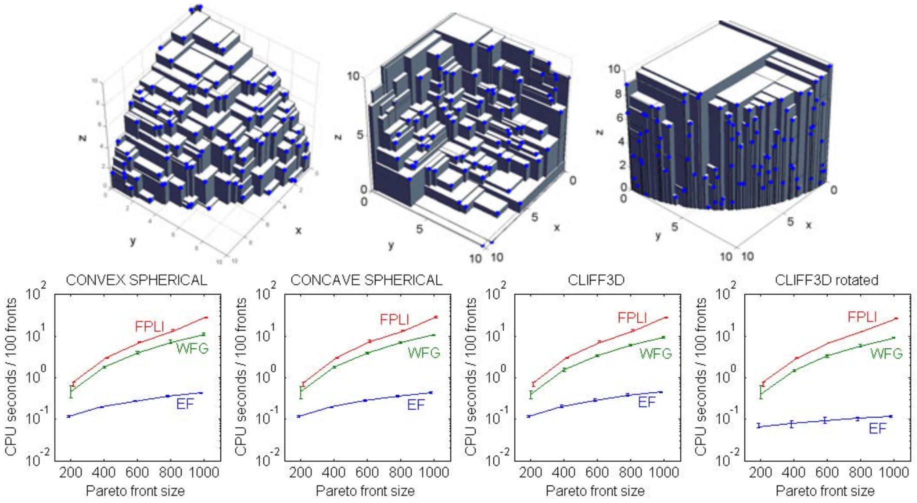


Fig. 4. Randomly generated fronts for the problems convexSpherical, concaveSpherical, and cliff3D with $n = 100$ (above, left to right) and speed tests on these fronts (below). For each type of front, 50 sets were generated. Error bars indicate mean, maximum, and minimum time consumption per algorithm and Pareto-front type and size.

final layer is reached. Therefore the time for processing this set is supposed to be high. The function `Clock()` is used to measure time in CPU seconds (measured using clock_t and time.h in MinGW/MS Windows). Compiler options are `g++ -O3` on a Toshiba Satellite PRO, Genuine Intel(R) CPU 1.66 GHz, T2300, with 1 GByte RAM. The tested algorithm implementations are:

EF: The dimension sweep algorithm discussed in this paper.[1]

FPLI: iterated application of total hypervolume computation with dimension sweep algorithm by Fonseca, Paquete and López-Ibáñez[2], cf. [12].

WFG: IHSO Algorithm[3] by Walking Fish Group (WFG) e.g. [6].

Figure 4 shows the results on the test-sets. It confirms the considerable speed gain of the new algorithm (EF) as compared to the other algorithms. Results are similar for different shapes of non-dominated fronts (Figure 4, concaveSpherical, convexSpherical, cliff3D). Swapping the $y-$ and $z-$ coordinates of the cliff3D problem (Figure 4, right, below) yields a further speed gain.

6 Conclusion and Outlook

The complexity of computing all contributions to the hypervolume given an approximation set $\mathrm{Y} \in \mathbb{A}^d$ has shown to be $\Theta(n \log n)$ when $d = 2$ and $d = 3$, and a lower bound of $\Omega(n \log n)$ has been established for $d > 3$. Also, the problem

[1] C++ source code is available from the authors on request.

[2] http://iridia.ulb.ac.be/$\sim$manuel/hypervolume

[3] http://www.wfg.csse.uwa.edu.au/toolkit/

of computing the hypervolume contribution of a single point has been considered, and has been shown to have complexity $\Theta(n)$ for $d = 2$, $\Theta(n \log n)$ for $d = 3$, and to be bounded by $\Omega(n \log n)$ for $d > 3$. An interesting aspect is that computing a single contribution has the same complexity in the 3-D case as computing all contributions, while in 2-D computing one contribution is less complex than computing all contributions.

A novel dimension sweep algorithm with asymptotically optimal time complexity $\mathcal{O}(n \log n)$ and space complexity $\mathcal{O}(n)$ for computing all hypervolume contributions in 3-D has been introduced. It improves existing algorithms [9] for this problem by a factor of $\sqrt{n}$. Empirical performance tests on randomly generated 3-D non-dominated fronts indicate that the new algorithm is considerably faster than existing algorithm implementations.

It is promising to apply the new algorithm in hypervolume-based archivers (e.g. [16]) and evolutionary algorithms (e.g. [20]), as it will make larger population/archive sizes and a higher number of iterations affordable. Interesting directions for future research could be the extension of the approach to problems in higher dimensions or to more general hypervolume-based subset selection problems, and the analysis of incremental update schemes.

Acknowledgments. Michael Emmerich gratefully acknowledges support by Fundação para a Ciência e a Technologia (FCT), Portugal. Grant: Set-Indicator Based Multiobjective Optimization (SIMO), SFRH/BPD/65330/2009, with funds from POPH of QREN Portugal 2007-2013 and the MCTES state budget. The authors would like to acknowledge Andreia Guerreiro and the anonymous reviewers for their insightful feedback on the manuscript.

References

1. Adelson-Velskij, G., Landis, E.: An algorithm for the organization of information. Doklady Akad. Nauk SSSR 146, 263–266 (1962)
2. Bader, J., Zitzler, E.: HypE: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation (2010) (in press)
3. Beume, N.: S-metric calculation by considering dominated hypervolume as Klee's measure problem. Evolutionary Computation 17(4), 477–492 (2009)
4. Beume, N., Fonseca, C.M., López-Ibáñez, M., Paquete, L., Vahrenhold, J.: On the complexity of computing the hypervolume indicator. IEEE Transact. Evolutionary Computation 13(5), 1075–1082 (2009)
5. Bradstreet, L., Barone, L., While, L.: Updating exclusive hypervolume contributions cheaply. In: Conf. on Evolutionary Computation, pp. 538–544. IEEE Press, Los Alamitos (2009)
6. Bradstreet, L., While, L., Barone, L.: A fast incremental hypervolume algorithm. IEEE Transact. on Evolutionary Computation 12(6), 714–723 (2008)
7. Bringmann, K., Friedrich, T.: Approximating the volume of unions and intersections of high-dimensional geometric objects. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 436–447. Springer, Heidelberg (2008)
8. Bringmann, K., Friedrich, T.: Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 6–20. Springer, Heidelberg (2009)

9. Bringmann, K., Friedrich, T.: An efficient algorithm for computing hypervolume contributions. Evolutionary Computation 18(3), 383–402 (2010)
10. 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)
11. 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)
12. Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Conf. on Evolutionary Computation, pp. 1157–1163. IEEE Press, Los Alamitos (2006)
13. Huband, S., Hingston, P., While, L., Barone, L.: An evolution strategy with probabilistic mutation for multi-objective optimisation. In: Conf. on Evolutionary Computation, vol. 4, pp. 2284–2291. IEEE Press, Los Alamitos (2003)
14. Igel, C., Hansen, N., Roth, S.: Covariance matrix adaptation for multi-objective optimization. Evolutionary Computation 15(1), 1–28 (2007)
15. Ishibuchi, H., Tsukamoto, N., Sakane, Y., Nojima, Y.: Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In: GECCO 2010, pp. 527–534. ACM, USA (2010)
16. Knowles, J., Corne, D., Fleischer, M.: Bounded archiving using the Lebesgue measure. In: Conf. on Evolutionary Computation, pp. 2490–2497. IEEE Press, Los Alamitos (2003)
17. Knuth, D.: The Art of Computer Programming, vol. 3. Addison-Wesley, Reading (1998)
18. Kung, H.T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. J. ACM 22(4), 469–476 (1975)
19. Mostaghim, S., Branke, J., Schmeck, H.: Multi-objective particle swarm optimization on computer grids. In: GECCO 2007, pp. 869–875. ACM, New York (2007)
20. Naujoks, B., Beume, N., Emmerich, M.: Multi-objective optimisation using S-metric selection: Application to three-dimensional solution spaces, vol. 2 (2005)
21. Overmars, M.H., Yap, C.K.: New upper bounds in klee's measure problem. SIAM J. Comput. 20(6), 1034–1045 (1991)
22. Preparata, F.P., Shamos, M.I.: Computational Geometry. Springer, Heidelberg (1985)
23. While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. IEEE Transact. Evolutionary Computation 10(1), 29–38 (2006)
24. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, ETH Zurich, Switzerland (1999)
25. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
26. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
27. Zitzler, E., Thiele, L., Laumanns, M., Foneseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transact. on Evolutionary Computation 7(2), 117–132 (2003)

Preference-Driven Co-evolutionary Algorithms Show Promise for Many-Objective Optimisation

Robin C. Purshouse, Cezar Jalbă, and Peter J. Fleming

Department of Automatic Control & Systems Engineering, University of Sheffield,
Mappin Street, Sheffield, S1 3JD, UK
{r.purshouse,p.fleming}@sheffield.ac.uk
http://www.shef.ac.uk/acse

Abstract. The simultaneous optimisation of four or more conflicting objectives is now recognised as a challenge for evolutionary algorithms seeking to obtain full representations of trade-off surfaces for the purposes of a posteriori decision-making. Whilst there is evidence that some approaches can outperform both random search and standard Pareto-based methods, best-in-class algorithms have yet to be identified. We consider the concept of co-evolving a population of decision-maker preferences as a basis for determining the fitness of competing candidate solutions. The concept is realised using an existing co-evolutionary approach based on goal vectors. We compare this approach and a variant to three realistic alternatives, within a common optimiser framework. The empirical analysis follows current best practice in the field. As the number of objectives is increased, the preference-driven co-evolutionary approaches tend to outperform the alternatives, according to the hypervolume indicator, and so make a strong claim for further attention in many-objective studies.

Keywords: many-objective optimisation, co-evolution, comparative study.

1 Introduction

Evolutionary algorithms are a popular technique for multi-objective optimisation [6]: their population-based nature being particularly useful for approximating trade-off surfaces from which a decision-maker can then select a final preferred solution (so-called *a posteriori* decision-making). However this conclusion was founded broadly upon studies considering bi-objective problems (such as the seminal work by Zitzler and colleagues [26]). More recent analyses, such as [23,4], on problems with higher numbers of objectives have suggested that the ability of some multi-objective evolutionary frameworks to represent the true trade-off surface becomes increasingly poor, and in some instances may be worse (according to some measures of approximation set quality [28]) than random search. In particular, the *sweet-spot* [12] of algorithm parameter settings that produce good results may contract markedly [23].

Research into evolutionary algorithms for approximating trade-off surfaces in problems with four or more objectives (sometimes referred to as *many-objective*

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 136–150, 2011.

optimisation) is still in its infancy. Acknowledging the tendency for solutions to be classed as equivalent (i.e. incomparable) when using the classical Pareto dominance operator in many-objective spaces, suggested approaches are often based on supplementing or replacing this pair-wise comparison operator with other pair-wise operators (for example, based on aggregation functions) or operators based on set comparisons (for example, comparisons based on hypervolumes) [25]. Since not all objectives are necessarily in conflict across the search space [22], other approaches have attempted to exploit such landscapes to reduce the dimensionality of the problem by either removing partially redundant objectives (for example, in [3]) or decomposing the problem into partially separable objective subsets (for example, in [21]). More comprehensive reviews can be found in [16] and [17].

Preference-based approaches are known to be useful for generating trade-off surfaces in objective sub-spaces of interest to the decision-maker, where preferences are specified by the decision-maker either a priori or progressively during the search (see, for example, [10]). This is because Pareto-based comparisons can be adapted to be more specific and so offer more opportunity for solutions to be comparable. A potentially interesting concept for a posteriori approaches is to consider a *family* of decision-makers representing sets of progressively interesting areas to explore. Intuitively, such an approach offers the potential for solution comparability in many-objective spaces. The family of preferences could be co-evolved simultaneously with the population of candidate solutions, such that the preferences maintain their usefulness as the search progresses. The only existing example of this type of approach known to the authors is the paper by Lohn and colleagues [19], where preferences are operationalised as target (or goal) vectors in objective-space. Target vectors are also central to algorithms proposed by Hughes (for example [15]), but such targets are not co-evolved.

In [19], good performance was observed on a set of bi-objective problems. We now explore the potential for this type of approach for many-objective optimisation. We retain Lohn et al.'s target vector formulation of preferences, but other representations may also be possible (for example, by considering both goals and priorities [10]). For a comprehensive review of co-evolutionary approaches for multi-objective optimisation, see [18]. Similarly, a review of the use of preferences in multi-objective evolutionary algorithms can be found in [24].

The remainder of the paper is organised as follows. The algorithms under consideration are described in Section 2, together with the empirical framework for performance comparison. Results of the algorithm testing, as the number of objectives to be optimised is successively increased, are presented in Section 3. A discussion of the findings is offered in Section 4, and Section 5 concludes.

2 Method

2.1 Overview

The general framework employed is aimed at a tractable and, so far as is feasible, robust analysis of algorithm performance. The co-evolutionary approach is

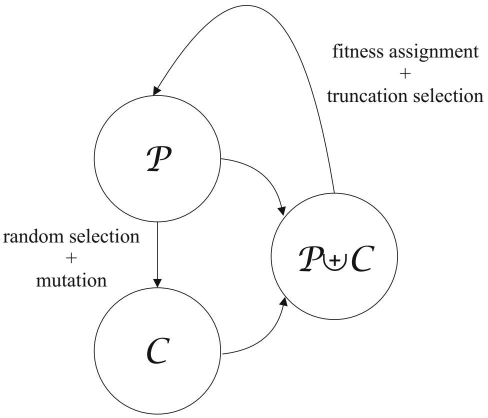


Fig. 1. Simple $(\mu + \lambda)$ framework used in the analysis

compared empirically to three potentially competitive methods, within a common overall algorithmic structure. The empirical testing broadly follows the best-practice approach developed in [27].

The common algorithmic framework is shown schematically in Fig. 1. It is a simple $(\mu+\lambda)$ elitist scheme. A population of candidate solutions, $\mathcal{P}$, of fixed size, N, is evolved for a fixed number of generations, t_{max}. N parents are randomly selected with replacement at iteration t from $\mathcal{P}(t)$ and are subjected to uniform mutation [6] to produce N children, $\mathcal{C}(t)$. $\mathcal{P}(t)$ and $\mathcal{C}(t)$ are then pooled and the combined population is sorted according to one of the methods under consideration. Truncation selection is then used to give the new parent population $\mathcal{P}(t+1)$. The co-evolutionary algorithms also each have a separate population of preferences that is described in more detail in Section 2.2.

We recognise that sacrifices have been made when using the above framework; in particular in terms of what levels of absolute performance might actually be achievable (for example, by including a recombination operator) or understanding how robust the findings are to algorithm parameter settings.

2.2 Methods for Ordering Candidate Solutions

Within the general framework described above, the different algorithms vary according to how the candidate solutions in $\mathcal{P}(t) \uplus \mathcal{C}(t)$ are ordered prior to truncation selection. Specific details of each ordering method (including a short label for reference purposes) are provided below.

Co-evolutionary: `coev_bin` and `coev_l2` In Lohn et al.'s co-evolutionary approach a symbiotic relationship is induced between a population of candidate solutions and (in our terms) a population of preferences [19]. Candidate solutions gain fitness by meeting a particular set of preferences, but the fitness contribution must be shared between other solutions that also satisfy those preferences (see (1) below). Preference sets only gain fitness by being satisfied by a candidate solution, but that fitness is reduced the more times the preferences are met by other solutions in the population (see (2) and (3) below). The overall aim is for the preferences to adaptively guide the solution population towards the Pareto optimal front, where the process of adaptation is determined in an evolutionary manner. When describing the algorithm, Kleeman et al. [18] used the analogy of greyhound racing, where the preference vectors act like the mechanical rabbit – keeping a certain distance from the dogs but never getting too far ahead or too close.

For our purposes, the algorithm presented in [19] is adapted to fit within the simple $(\mu+\lambda)$ framework. Following Lohn et al.'s notation, the preference vectors are known as *target objective vectors* (TOVs):

1. Set $t = 1$.
2. Initialise population of N candidate solutions $\mathcal{P}(t)$ and population of N_T TOVs $\mathcal{T}(t)$. Note that an objective range must be defined for the TOVs – typically based on estimates of the ideal and anti-ideal objective vectors.
3. Evaluate $\mathcal{P}(t)$ against the objectives for the problem at hand.
4. Apply random selection with replacement to $\mathcal{P}(t)$ followed by uniform mutation to produce N new candidate solutions $\mathcal{C}(t)$.
5. Evaluate $\mathcal{C}(t)$ against the objectives for the problem at hand.
6. Randomly generate a further TOV population, $\mathcal{U}(t)$, of size N_T.
7. Check every solution in $\mathcal{P}(t) \uplus \mathcal{C}(t)$ against every TOV in $\mathcal{T}(t) \uplus \mathcal{U}(t)$ and keep a record of which solutions satisfy which TOVs. Let $n_{\mathbf{t}}$ be the number of solutions that satisfy (ie. weakly dominate, denoted $\preceq$) TOV $\mathbf{t}$.
8. Score each solution, $\mathbf{p}$, in $\mathcal{P}(t) \uplus \mathcal{C}(t)$ according to:
$$s_{\mathbf{p}} = \sum_{\mathbf{t} \in \mathcal{T}(t) \uplus \mathcal{U}(t) | \mathbf{p} \preceq \mathbf{t}} \frac{1}{n_{\mathbf{t}}} \tag{1}$$
9. Score each TOV, $\mathbf{t}$, in $\mathcal{T}(t) \uplus \mathcal{U}(t)$ according to:
$$s_{\mathbf{t}} = \frac{1}{1+\alpha} \tag{2}$$
where:
$$\alpha = \begin{cases} 1 & \text{if } n_{\mathbf{t}} = 0, \\ \frac{n_{\mathbf{t}}-1}{2N-1} & \text{otherwise.} \end{cases} \tag{3}$$
10. Apply truncation selection to scored $\mathcal{P}(t) \uplus \mathcal{C}(t)$ to get $\mathcal{P}(t+1)$ of size N.
11. Apply truncation selection to scored $\mathcal{T}(t) \uplus \mathcal{U}(t)$ to get $\mathcal{T}(t+1)$ of size N_T.
12. Increment t and if $t < t_{max}$ then return to step 4. Otherwise terminate.

As a simple example to illustrate the workings of the algorithm, consider a bi-objective minimisation case with two solution vectors, $\mathbf{p}_1$ and $\mathbf{p}_2$, and two TOVs, $\mathbf{t}_1$ and $\mathbf{t}_2$ shown in Fig. 2. In the example, $N = 1$ (and $N_T = 1$) since scoring is done on the combined parent and child populations. $\mathbf{t}_1$ is satisfied by

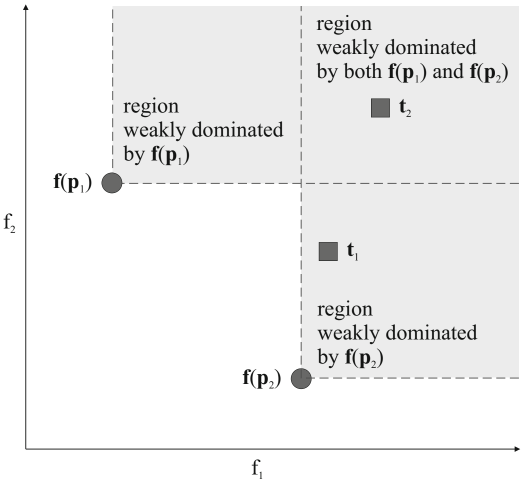


Fig. 2. Simple bi-objective minimisation example to illustrate algorithm `coev_bin`

$\mathbf{p}_2$ only and therefore $n_{\mathbf{t}_1} = 1$. $\mathbf{t}_2$ is satisfied by both solutions and so $n_{\mathbf{t}_2} = 2$. In terms of scoring the solutions, from step 8 $s_{\mathbf{p}_1} = 1/n_{\mathbf{t}_2} = 1/2$ whilst $s_{\mathbf{p}_2} = 1/n_{\mathbf{t}_1} + 1/n_{\mathbf{t}_2} = 3/2$. In terms of scoring the TOVs, from step 9 α for $\mathbf{t}_1$ is $(n_{\mathbf{t}_1} - 1)/(2N - 1) = (1 - 1)/(2 \times 1 - 1) = 0$ and therefore $s_{\mathbf{t}_1} = 1$. Similarly α for $\mathbf{t}_2$ is $(2 - 1)/(2 \times 1 - 1) = 1$ and so $s_{\mathbf{t}_2} = 1/2$.

The scoring system in the above algorithm, denoted `coev_bin`, is based on the weak Pareto dominance relation, producing a binary outcome for the success of a candidate solution in meeting any TOV. Intuitively, this approach may also suffer from incomparability problems as the number of objectives increases. Therefore an extension of the algorithm is also proposed, in which the all-or-nothing weak dominance relation is replaced with a Euclidean distance measure. Distances are normalised over the length of the vector from the ideal to the anti-ideal point. This variant is denoted `coev_l2`. In stages 8 and 9 of the original algorithm, $n_{\mathbf{t}}$ is replaced by the inverse of the normalised Euclidean distance between the solution and the TOV in cases where the TOV is not met, subject to a neighbourhood constraint determined as per conventional niche size selection (see p155 of [6]).

Dominance and density: `nds_c` The familiar non-dominated sorting and crowding scheme from NSGA-II is used [5]. Methods of this type are known to perform well on bi-objective problems but may experience difficulties in many-objective spaces [26,23,17].

Average ranking and sharing: `ar_epa` Corne & Knowles [4] found that a simple approach based on taking the mean ranking of each candidate solution against each objective [1] performed well in relation to other suggested many-objective approaches, according to the coverage metric, on a selection of multi-objective travelling salesman and single-machine job-shop problems in the presence of many objectives and high degrees of conflict.

In this study, average ranking is combined with a sharing scheme based on the Epanechnikov kernel according to the method given in [8].

Random: `rand` Evidence exists that random search can be competitive to evolutionary approaches in many-objective spaces [23,4], making this a natural benchmark, at present, for comparison against any proposed new algorithm.

This study implements a very crude random scheme: $N \times t_{max}$ candidate solutions are randomly generated, dominated solutions are removed and a maximum of N solutions are then drawn at random without replacement as a representative final population for comparison purposes.

2.3 Test Problems, Performance Indicators and Parameter Settings

The different algorithms are tested against the even-numbered problems from the *WFG* test suite [14] for 2, 7 and 13-objective instances of the problems. In all cases, the WFG parameters k and l are set to 12 and 20 respectively, providing a constant number of decision variables for each problem instance. Problem attributes covered include: separability versus non-separability; unimodality versus multimodality; unbiased versus biased parameters; and convex versus concave geometries.

A sequence of performance indicators has been chosen according to the approach in [27]. Firstly, comparisons are made in terms of whether one approximation set dominates another, since the only preference information this metric relies on is a direction of preference in each objective. Where such limited preference information is unable to distinguish between algorithms, approximation set comparisons are then made using the hypervolume metric (which assumes equal relative importance of normalised objectives across the search domain). The hypervolume is calculated using the method and software developed by Fonseca and colleagues [11]. Note that visualisation of approximation sets has only been attempted for the 2-objective instances of the problems. Median attainment surfaces [9] are provided.

The parameter settings shown in Table 1 are held constant across all algorithm runs. Ideally we would wish to consider how the parameter sweet spots are affected by objective-space dimension (and in particular to ensure we have avoided bias toward or against a particular algorithm) but such testing is beyond the scope of the current study.

2.4 Statistical Treatment

Performance comparisons between algorithms are made according to a rigorous non-parametric statistical framework, drawing on recommendations in [27]. 31 initial populations of candidate solutions are generated for each WFG problem instance. The competing algorithms are then evolved on each of these initial populations using fresh random number seeds. The approximation sets used in

Table 1. Algorithm parameter settings: note that the mutation rate, p_m, is applied at the phenotypic level; the target objective vector population size, N_T, is relevant to coev_bin and coev_l2 only

Parameter	Value
Candidate solution population size, N	100
Target objective vector population size, N_T	100
Generations, t_{max}	250
Mutation rate, p_m	$1/(k+l) = 1/32$

the comparisons are the non-dominated solutions in the final population of each algorithm run.

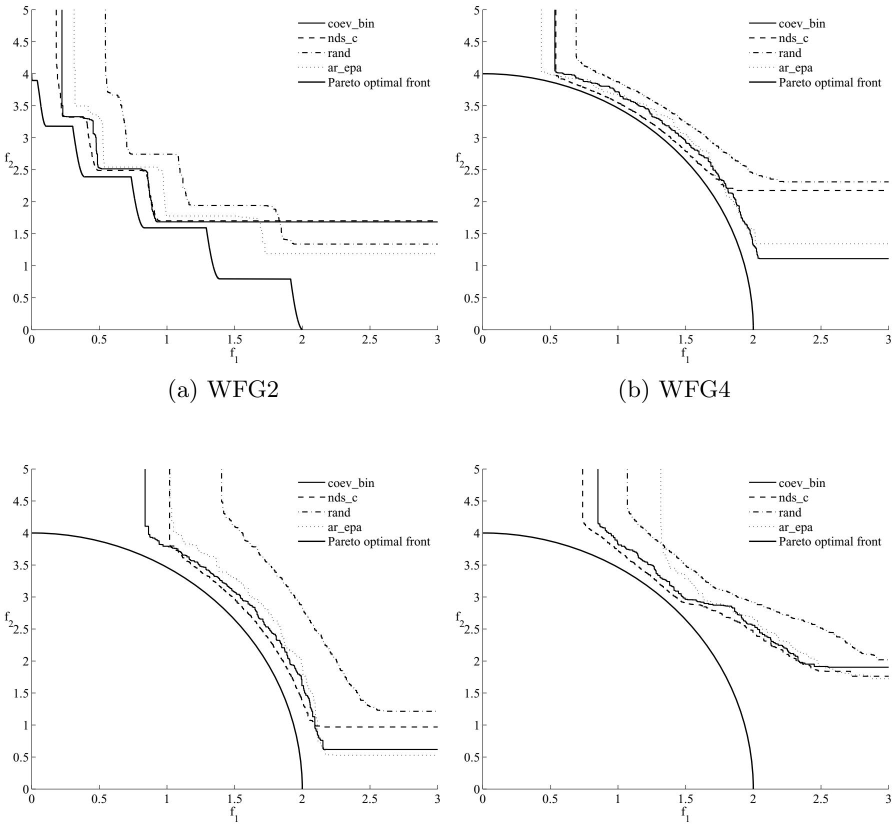
Comparisons based on dominance rank. A purely pair-wise approach is used. The chosen statistical metric is the difference between the mean dominance rank of algorithm_A against the mean dominance rank of algorithm_B over the pooled 62 runs of the two algorithms on a particular problem instance. Randomisation testing [20] is then performed where the algorithm labels for the pair of runs related to each initial population are subject to random switching prior to recalculation of the statistical metric. This process is repeated 999 times and the metrics are sorted. With Bonferroni correction [27], we require an observed result at location 1, 2, 999, or 1000 in the resulting distribution to achieve a statistically significant difference between algorithm_A and algorithm_B at the 5%-level.

Comparisons based on hypervolume. For each problem instance we effectively have a randomised complete block design, where for each of our five *treatments* (algorithms) we have one *observation* (hypervolume) for each of 31 *blocks* (initial populations). Following the approach in [13], we first test the hypothesis that all algorithms perform equally by computing the large-sample approximation for the Friedman statistic, S. If this hypothesis is rejected at the 5%-level, we then consider pair-wise comparisons between the algorithms using the large-sample approximation to the Wilcoxon-Nemenyi-McDonald-Thompson two-sided all-treatments multiple comparison procedure at an experimentwise error rate of 1%.

3 Results

3.1 Attainment Surface Results

Considering as a starting point the bi-objective results, plots of median attainment surfaces across the 31 runs of each algorithm on each test problem are shown in Fig. 3. These allow visual inspection of performance in terms of the dual aims of proximity to and diversity across the global trade-off surface [2]. In general, all methods exhibit issues with diversity across the test problems. This may in part



(a) WFG2

(b) WFG4

(c) WFG6

(d) WFG8

Fig. 3. Attainment surfaces for 2-objective instances of the even-numbered WFG test problems - the results for `coev_l2` are very similar to `coev_bin` and, for the purposes of clarity, are therefore not shown

be due to the very simple variation operator used, since existing results for NSGA-II (an `nds_c` algorithm with more refined recombination and mutation operators) tend to show better diversity on the WFG problems [14]. Across the test problems, proximity is observed to be tightest on the separable WFG4.

On the whole, in terms of visual inspection, there is little to choose between the evolutionary methods from the perspective of median attainment surfaces. Random search is, in general, clearly the worst performer, although it does achieve improved coverage of the bottom-right portion of the trade-off surface for WFG2 in comparison to `nds_c` and `coev_bin`. Of the competing methods, the classic dominance and density-based approach (`nds_c`) provides the best proximity but its diversity is sometimes relatively lacking. All methods are unable to provide a good representation of the bottom half of the trade-off surface for WFG8. In this problem, distance-related parameters are dependent on position-related parameters and — as Huband et al. previously demonstrated using NSGA-II [14] — even the bi-objective instance represents a challenge for evolutionary optimisers.

3.2 Dominance Rank Results

Results of the randomisation testing based on the dominance rank metric are shown in Table 2. We follow the approach in [4] where a partial ordering of algorithms is constructed based on the notion of **all** algorithms presented on a higher row of the table, within the context of a test problem instance, outperforming (according to the procedures in Section 2.4) those presented on a lower row of the table. Order of presentation within a row is based purely on the order in which the algorithms were introduced in Section 2.2. Any isolated cases of one algorithm outperforming another (but where a partial ordering considering all algorithms cannot be constructed) are described separately in the text.

The dominance rank metric, which requires the weakest assumptions about decision-maker preferences, is unable to provide any discrimination between the algorithms for any of the 7 or 13-objective test problems. At the 2-objective level, random search was outperformed by the four other algorithms on WFG2 and WFG6. Drilling into the pair-wise comparisons at the 2-objective level, `ar_epa` outperformed `rand` on WFG2 and `nds_c` outperformed all except `coev_bin` on WFG8. `ar_epa` was unable to beat `rand` on this problem, which prevented any partial ordering of algorithms.

Table 2. Dominance rank results

WFG#	# objectives	Ranking by performance	
2	2	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
	7	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
	13	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`

4	2	1st
		2nd
			`coev_bin, coev_l2, nds_c, ar_epa`
			`rand`

	7	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
	13	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`

6	2	1st
		2nd
			`coev_bin, coev_l2, nds_c, ar_epa`
			`rand`

	7	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
	13	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
8	2	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
	7	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`
	13	1st	`coev_bin, coev_l2, nds_c, ar_epa, rand`

Table 3. Hypervolume results

WFG#	# objectives	Ranking by performance	
2	2	1st	coev_bin, coev_l2, nds_c, ar_epa
		2nd	rand
	7	1st	coev_bin, coev_l2
		2nd	nds_c, ar_epa
		3rd	rand
	13	1st	coev_bin, coev_l2
		2nd	nds_c, ar_epa
		3rd	rand
4	2	1st	coev_bin, coev_l2, ar_epa
		2nd	nds_c
		3rd	rand
	7	1st	coev_bin, coev_l2
		2nd	ar_epa
		3rd	nds_c
		4th	rand
	13	1st	coev_bin, coev_l2, ar_epa
		2nd	nds_c
		3rd	rand
6	2	1st	coev_bin, coev_l2, nds_c, ar_epa
		2nd	rand
	7	1st	coev_bin, coev_l2
		2nd	ar_epa
		3rd	nds_c
		4th	rand
	13	1st	coev_bin, coev_l2, ar_epa
		2nd	nds_c
		3rd	rand
8	2	1st	nds_c
		2nd	coev_bin, coev_l2
		3rd	ar_epa
		4th	rand
	7	1st	coev_bin, coev_l2
		2nd	ar_epa
		3rd	nds_c, rand
	13	1st	coev_bin, coev_l2
		2nd	ar_epa
		3rd	rand
		4th	nds_c

3.3 Hypervolume Results

Using the stronger preference assumptions implicit in the hypervolume metric, more interesting things can be said about the many-objective performance of the algorithms under test. The initial Friedman test breaks the hypothesis that all five algorithms are equivalent, for every problem instance, with S-values ranging between 74 for 2-objective WFG6 and 115 for 2-objective WFG8. Therefore pairwise statistical comparisons can now be considered, the outcomes of which are shown in Table 3.

On the 2-objective problems, all four evolutionary methods outperform random search. The four algorithms are otherwise equivalent on WFG2 and WFG6, with the three specifically many-objective algorithms outperforming the classical approach on WFG4 but being outperformed themselves on WFG8.

Moving into the many-objective domain of 7-objective problem instances, the two co-evolutionary algorithms now consistently outperform `ar_epa` and `nds_c`. `ar_epa` also outperforms `nds_c` on three of the four problems. Also the latter classical approach is now indistinguishable from random search on WFG8, despite its earlier high performance on the 2-objective version of this problem.

As the number of objectives is now increased to 13, `ar_epa` reasserts its competitiveness with the coevolutionary algorithms. `nds_c` is now indistinguishable from random search on WFG4 and is actually performing worse than random search on WFG8.

4 Discussion

4.1 Findings

This new empirical study of the many-objective behaviour of multi-objective evolutionary algorithms provides further confirmatory evidence of how relative performance, attributable to particular algorithm components, changes as the number of conflicting objectives to be optimised is increased:

- Classical components based on dominance and density measures can perform well on bi-objective problems in a variety of domains, but performance (as measured by various approximation set metrics) may degenerate to random search – or worse – as the number of objectives is increased.
- A simple approach based on average ranking, possibly supplemented by a mechanism to promote objective-space diversity, tends to outperform (subject to arguably reasonable preference assumptions) both standard methods and random search on many-objective problems (at least within realistic constraints on the number of objectives that would perhaps be faced in practice – up to 20 in the study by Corne & Knowles [4]).

Our study has also identified that the concept of co-evolving a family of preferences, and using this population to evaluate candidate solutions, shows real promise for many-objective optimisation. Within the settings considered, the methods tend to remain competitive with standard approaches in bi-objective spaces and are also found to be consistently among the best choices for higher numbers of objectives. In our implementations of the co-evolutionary concept, the relaxation of the all-or-nothing scoring criterion appeared to offer little beyond the weak Pareto dominance approach originally used by Lohn and colleagues [19].

There are some curiosities in the observed results in terms of the relative performance of the algorithms as the number of objectives is increased. On the non-separable biased-parameter WFG8, `nds_c` outperforms the other methods (in terms of hypervolume), but by 7 objectives it is no longer distinguishable from `rand` and by 13 objectives it is now worse than `rand` — the only occasion where random search outperforms an evolutionary algorithm in the study. There is some existing evidence, collected on the *DTLZ2* test problem [7], that active diversity operators may lead to worse results than random search for `nds_c`-type algorithms due to favouring of solutions that have very poor proximity in absolute terms but which are non-dominated with respect to their peers [23]. The size of dominated objective-space, relative to the scaling of the global trade-off surface, is substantially smaller in the WFG problems than for the DTLZ2 problem which might to some extent mitigate the possibility of this type of result; yet perhaps the inter-dependency between proximity and diversity on WFG8 provides just enough difficulty to recreate these conditions.

Across all four bi-objective test problems, equivalent performance is observed between `ar_epa` and the two co-evolutionary algorithms. By 7 objectives `ar_epa` is now being consistently outperformed, but interestingly by 13 objectives the algorithm has regained its equivalence on two of the four test problems. Since our comparisons are made purely in relative terms, this may actually represent more of a deterioration in the absolute performance of `coev_bin` and `coev_l2` than any improvement in the performance of the average ranking method, perhaps suggesting that all algorithms will collapse to equivalent performance given a sufficiently large number of objectives. Regardless, this finding suggests a degree of complexity in how algorithm performance decays with objective dimensionality.

4.2 Limitations and Areas for Future Research

The study has two main limitations. The first is that the algorithm framework studied is less complex than a typical multi-objective evolutionary algorithm, particularly in terms of variation operator refinement and archiving strategy. Therefore caution is advised in generalising the results, although it should be noted that findings for `nds_c`, `ar_epa` and `rand` for the most part reflect the existing evidence for these types of operators when housed in a variety of algorithm configurations and solving a variety of test problems [4,23,17]. The second main limitation of the study is the absence of analysis of how performance varies with the tuneable parameters of the algorithms (for our study these are essentially those shown in Table 1). This type of analysis can be very computationally demanding to perform but is important for gaining insight into the robustness of the algorithms and to mitigate the possibilities of producing misleading findings due to inadvertent ‘tuning to the problem’. A final minor point to consider is potential confounding of the effects of objective-space dimension with other factors. In the standard form of the WFG test suite [14], when we increase the number of objectives we have the choice of either explicitly increasing the number of decision variables or implicitly collapsing the scale of the reduction transformations. We chose the latter but a better mitigation might have been to consider simultaneously the effects of both options.

In terms of further research, the goal vector approach examined in this paper represents just one possible formulation of preference and further research may be worthwhile looking at other realisations of the concept. Consideration may also be given to scaling up the empirical test framework to include other existing many-objective algorithms and, crucially, systematic analyses of sweet-spots.

5 Conclusion

The field of evolutionary multi-objective optimisation has developed rapidly over the past 10–15 years, yet the design of effective algorithms for addressing problems with more than three objectives remains a key challenge. The realisation that random search is often a competitive solution technique in these situations makes clear the scale of the task, and a major breakthrough continues to elude researchers. However such many-objective optimisation problems will continue to arise in real-world problems and the need for efficient methodologies is pressing.

This study is the first of its kind to explore the concept of co-evolving a family of preferences as a basis for discovering better quality trade-off surfaces for many-objective problems. To establish whether this concept holds promise and deserves further investigation, it has been compared to other potentially competitive methods using a tractable and rigorous empirical framework.

The major outcome of this study is the revelation that the co-evolutionary approach shows evidence of good performance, and can therefore make a strong claim for inclusion as a comparator in future analyses of algorithms for many-objective optimisation.

References

1. Bentley, P.J., Wakefield, J.P.: Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In: Chawdhry, P.K., Roy, R., Pant, R.K. (eds.) Soft Computing in Engineering Design and Manufacturing, pp. 231–240. Springer, Heidelberg (1997)
2. Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(2), 174–188 (2003)
3. Brockhoff, D., Saxena, D.K., Deb, K., Zitzler, E.: On handling a large number of objectives a posteriori and during optimization. In: Knowles, J., Corne, D., Deb, K. (eds.) Multiobjective Problem Solving from Nature. Natural Computing Series, pp. 377–403. Springer, Heidelberg (2008)
4. Corne, D., Knowles, J.: Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In: Proceedings of GECCO 2007, pp. 773–780 (2007)

5. 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)
6. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
7. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation, vol. 1, pp. 825–830 (2002)
8. Fonseca, C.M., Fleming, P.J.: Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction. In: Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, pp. 42–52 (1995)
9. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 584–593. Springer, Heidelberg (1996)
10. Fonseca, C.M., Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms — part I: A unified formulation. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28(1), 26–37 (1998)
11. Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Proceedings of the 2006 Congress on Evolutionary Computation, pp. 1157–1163 (2006)
12. Goldberg, D.: The race, the hurdle and the sweet spot. In: Bentley, P. (ed.) Evolutionary Design by Computers, pp. 105–118. Morgan Kaufmann, San Francisco (1999)
13. Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods, 2nd edn. Wiley, New York (1999)
14. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477–506 (2006)
15. Hughes, E.J.: MSOPS-II: A general-purpose many-objective optimiser. In: Proceedings of the 2007 Congress on Evolutionary Computation, pp. 3944–3951 (2007)
16. Hughes, E.J.: Fitness assignment methods for many-objective problems. In: Knowles, J., Corne, D., Deb, K. (eds.) Multiobjective Problem Solving from Nature. Natural Computing Series, pp. 307–329. Springer, Heidelberg (2008)
17. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: Proceedings of the 2008 Congress on Evolutionary Computation, pp. 2419–2426 (2008)
18. Kleeman, M.P., Lamont, G.B.: Coevolutionary multi-objective EAs: The next frontier? In: Proceedings of the 2006 Congress on Evolutionary Computation, pp. 1726–1735 (2006)
19. Lohn, J.D., Kraus, W.F., Haith, G.L.: Comparing a coevolutionary genetic algorithm for multiobjective optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, pp. 1157–1162 (2002)
20. Manly, B.F.J.: Randomization and Monte Carlo Methods in Biology. Chapman and Hall, Boca Raton (1991)
21. Purshouse, R.C., Fleming, P.J.: An adaptive divide-and-conquer methodology for evolutionary multi-criterion optimisation. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 133–147. Springer, Heidelberg (2003)

22. Purshouse, R.C., Fleming, P.J.: Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 16–30. Springer, Heidelberg (2003)
23. Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. IEEE Transactions on Evolutionary Computation 11(6), 770–784 (2007)
24. Rachmawati, L., Srinivasan, D.: Preference incorporation in multi-objective evolutionary algorithms: A survey. In: Proceedings of the 2006 Congress on Evolutionary Computation, pp. 962–968 (2006)
25. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007)
26. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)
27. Zitzler, E., Knowles, J., Thiele, L.: Quality assessment of pareto set approximations. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 373–404. Springer, Heidelberg (2008)
28. 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)

Adaptive Objective Space Partitioning Using Conflict Information for Many-Objective Optimization

Antonio López Jaimes[1], Carlos A. Coello Coello[1],
Hernán Aguirre[2], and Kiyoshi Tanaka[2]

[1] CINVESTAV-IPN, Computer Science Department, Mexico City 07360, Mexico
tonio.jaimes@gmail.com, ccoello@cs.cinvestav.mx
[2] Shinshu University, Faculty of Engineering, Nagano 380-8553, Japan
{ahernan,ktanaka}@shinshu-u.ac.jp

Abstract. In a previous work we proposed a scheme for partitioning the objective space using the conflict information of the current Pareto front approximation found by an underlying multi-objective evolutionary algorithm. Since that scheme introduced additional parameters that have to be set by the user, in this paper we propose important modifications in order to automatically set those parameters. Such parameters control the number of solutions devoted to explore each objective subspace, and the number of generations to create a new partition. Our experimental results show that the new adaptive scheme performs as good as the non-adaptive scheme, and in some cases it outperforms the original scheme.

1 Introduction

Multi-objective Evolutionary Algorithms (MOEAs) have been successfully applied to solve many real world multi-objective problems (MOPs) (e.g.,[1]). However, recent experimental [2,3,4] and analytical [5,6] studies have pointed out that MOEAs based on Pareto optimality have some drawbacks to solve problems with a large number of objectives (these problems are usually called *many-objective problems*). Approaches to deal with such problems have mainly focused on the use of alternative optimality relations [7,8], reduction of the number of objectives of the problem, either during the search process [9,10] or, at the decision making process [11,12,13], and the incorporation of preference information [4].

A general scheme for partitioning the objective space in several subspaces in order to deal with many-objective problems was introduced in [14]. In that approach, the solution ranking and parent selection are independently performed in each subspace to emphasize the search within smaller regions of objective function space. Later, in [15] we proposed a new partition strategy that creates objective subspaces based on the analysis of the conflict information. By grouping objectives in terms of the conflict among them, we aim to divide the MOP into several subproblems in such a way that each subproblem contains the information to preserve as much as possible the structure of the original problem. Since the conflict among objectives may change along the search space, the

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 151–165, 2011.

search on the subspaces should be alternated with a search on the entire objective space in order to update the conflict information. The partitioning approach is more closely related to objective reduction approaches, specially those adopted during the search (e.g., [11,9,10]). However, its main difference with respect to those approaches is that the partitioning scheme incorporates all the objectives in order to cover the entire Pareto front.

Although the partitioning strategy based on conflict information produced good results compared with NSGA-II and another partitioning strategy, the conflict-based strategy introduced some additional parameters whose values need to be defined by the user. These parameters are *i*) the number of generations to search on the partition's subspaces and on the whole objective space, and *ii*) the number of solutions assigned to explore each subspace. This represents a problem since the best values for those parameters may depend on the problem to be solved. Additionally, some subspaces have greater degree of conflict than others. Therefore, it seems reasonable to put more emphasis on them by assigning them more resources (in our case, a larger population).

In this paper we present important modifications in order to automate the determination of those parameters and to take advantage of the conflict information to focus the search on some subspaces. In particular, the conflict information is employed to assign the number of solutions according to the contribution of each subspace to the total conflict of the problem. To automatically define the parameter value described in *ii*), we use a convergence detection method based on two statistical hypothesis tests applied on the values of a quality indicator.

The remainder of the paper is organized as follows. Section 2 presents the main concepts and notation used through out the paper. The basic objective space partitioning framework is introduced in Section 3. The motivation and details of the automation of the partitioning parameters are described in Section 4. In Section 5 is presented the experimental evaluation of the adaptive scheme. Finally, in Section 6 we draw some conclusions about the new proposed scheme, as well as some future research paths.

2 Basic Concepts and Notation

Definition 1 (Objective space $\varPhi$). *The objective space of a MOP is the set $\varPhi = \{f_1, f_2, \ldots, f_M\}$ of the M objective functions to be optimized.*

Definition 2 (Subspace ψ). *A subspace ψ of $\varPhi$ is a lower dimensional space that includes some of the objective functions in $\varPhi$, i.e. $\psi \subset \varPhi$.*

Definition 3 (Space partition $\varPsi$). *A space $\varPhi$ is said to be partitioned into N_S subspaces, denoted as $\varPsi$, if $\varPsi = \{\psi_1, \psi_2, \ldots, \psi_{N_S} \,|\, \cup_{i=1}^{N_S} \psi_i = \varPhi \wedge \cap_{i=1}^{N_S} \psi_i = \emptyset\}$.*

Definition 4 (Pareto dominance relation). *A solution $\mathbf{x}^1$ is said to Pareto dominate solution $\mathbf{x}^2$ in the objective space $\varPhi$, denoted by $\mathbf{x}^1 \prec \mathbf{x}^2$, if and only if (assuming minimization): $\forall f_i \in \varPhi : f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \wedge \exists f_i \in \varPhi : f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$.*

Definition 5 (Pareto optimal set). *The Pareto optimal set,* P_{opt}*, is defined as:* $P_{\text{opt}} = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{y} \in \mathcal{X} : \mathbf{y} \prec \mathbf{x}\}$*, where* $\mathcal{X} \subseteq \mathbb{R}^n$ *is the feasible set in decision variable space.*

Definition 6 (Pareto front). *For a Pareto optimal set* P_{opt}*, the Pareto front,* PF_{opt}*, is defined as:* $PF_{\text{opt}} = \{\mathbf{z} = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \mid \mathbf{x} \in P_{\text{opt}}\}$*. We will denote by* PF_{approx} *the Pareto front approximation achieved by a MOEA.*

Definition 7 (Pareto front approximation). *A Pareto front approximation, denoted by* PF_{approx}*, is a finite set composed of nondominated solutions.*

Of course, in practice the goal of a MOEA is finding a PF_{approx} with the best quality from all the possible approximation sets that could be generated. The quality is usually defined in terms of both convergence and spread [16].

3 The Conflict-Based Partitioning Framework

3.1 General Idea of the Partitioning Framework

The basic idea of the partitioning framework is to divide the objective space into several subspaces so that a different portion of the population focuses the search in a different subspace. By partitioning the objective space into subspaces, we aim to emphasize the search within smaller regions of objective space. Instead of dividing the population into independent subpopulations, a fraction of the pool of parents for the next generation is selected based on a different subspace. This way, the pool of parents will be composed with individuals having a good performance in each subspace. In our approach, we partition the M-dimensional space $\Phi = \{f_1, f_2, \ldots, f_M\}$ into N_S non-overlapping subspaces $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_S}\}$. We selected NSGA-II to implement our proposed partitioning framework. Thus, the nondominated sorting and truncation procedures of NSGA-II are modified in the following way. The union of the parents and offspring, $\mathcal{P} \cup \mathcal{Q}$, is sorted N_S times using a different subspace each time. Then, from each mixed sorted population, the best $|\mathcal{P}|/N_S$ solutions are selected to form a new parent population of size $|\mathcal{P}|$. After this, the new population is generated by means of recombination and mutation using binary tournaments.

3.2 Using Conflict Information to Partition the Objective Space

The number of all possible ways to partition Φ into N_S subspaces is very large. Therefore, it is not feasible to search in all the possible subspaces. Instead, we can define a schedule of subspace sampling by using a partition strategy. In [14], three strategies to partition Φ were investigated: random, fixed, and shift partition. Later, we proposed a new strategy using the conflict information among objectives [10]. In that strategy the first partition contains the least conflicting objectives, the second one the next least conflicting objectives, and so on. Therefore, instead of removing the least conflicting objectives, those objectives are integrated to form subspaces in such a way that all the objectives are optimized. By grouping objectives in terms of the conflict among them, we are trying to separate the MOP into subproblems in such a way that each subspace contains information to preserve most of the structure of the original problem. The correlation among solutions in PF_{approx} is defined to estimate the conflict among objectives in the sense defined by Carlson and Fullér [17]. A negative correlation between a pair of objectives means that one objective increases while the other decreases and vice versa. Thus, a negative correlation estimates the conflict between a pair of objectives. On the other hand, if the correlation is positive, then both objectives increase or decrease at the same time. That is, the objectives support each other.

In order to implement the new partition strategy we should take into account two issues: *i*) the conflict relation among the objectives may change during the search, and *ii*) the tradeoffs between objectives in different subspaces are not taken into account. In a sense, the vector evaluated algorithm (VEGA) [18] can be thought of as a particular case of the partitioning framework in which each objective is a partition. The issue *ii*) is related with the "speciation" phenomena observed in VEGA in which some individuals excel in some objectives (some subspaces in our case). To deal with both issues, in the final partitioning framework the search is divided in several stages. Each of these stages is divided in two phases, namely, an approximation phase followed by a partitioning phase. In the approximation phase all the objectives are used to select the new parent population (we can view this as a partition with a single partition formed by all the objectives). The goal of this phase is finding a good approximation of the current PF_{opt}. The interested reader is referred to [15] to find the details of the entire algorithm.

4 Automatic Setting of the Partitioning Parameters

As mentioned in the previous section, the conflict-based partitioning scheme needs the definition of some important parameter values. That is, the number of generations assigned to each phase of the strategy, and the number of total stages during the search. In this section we will introduce some modifications aimed to eliminate the need to define those values manually. In addition, as we will see in this section, some subspaces have greater conflict contribution than others. Therefore, it seems reasonable to put more emphasis on them by assigning them more solutions. In a similar way, in some subspaces the search could stagnate earlier than in others. Therefore it could be a good idea to stop searching them and reassign those resources to other subspaces where progress can still continue.

4.1 Proportional Assignment of the Resources

One of the findings of our previous work was the fact that the conflict among certain objectives is considerably larger than the conflict among others. In order to measure the contribution of each subspace to the total conflict in the problem, we compute for each subspace its "conflict degree", i.e., the sum of the conflict between each pair of objectives. The ratio of the conflict degree of each subspace and the total conflict is called the *conflict contribution*. Figure 1 presents the conflict contribution of 3 subspaces through the optimization process in the Knapsack problem. In the figure we can clearly see that subspace 3 has a larger conflict contribution with respect to the other subspaces. In the original strategy proposed in [15], each subspace receives an equal number of parents regardless its conflict contribution. However, we can take advantage of the conflict information in order to distribute the number of parents accordingly. That is, the proportion of parents granted for each subspace should be proportional to its contribution to the total conflict on the problem. In order to illustrate this idea, let's take as an example the conflict contribution obtained for each of the 3 subspaces in the Knapsack problem (Fig. 2).

Since subspace 3 contributes with 40% of the total conflict, then it will receive that percentage of the total number of parents. In turn, subspaces 1 and 2 will

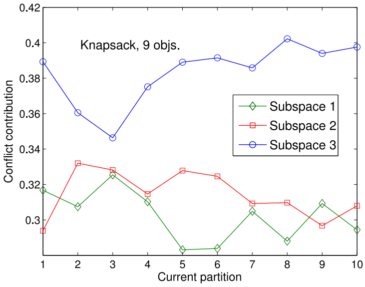


Fig. 1. Conflict contribution of three subspaces during the search process

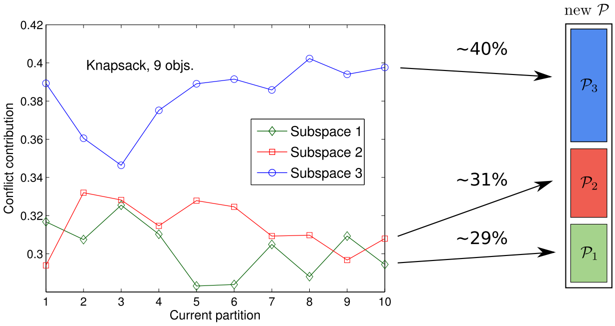


Fig. 2. Proportional assignment of parents according to the conflict contribution of the subspaces

receive 29% and 31% of the parents, respectively. The motivation behind this idea is the fact that the most conflicting objectives are the ones that change the most the Pareto optimal front when they are removed [12]. Therefore, they contribute with a larger trade-off surface to form the whole Pareto front. Thus, the most conflicting subspace should have more solutions in order to cover larger areas of the Pareto front. The modified procedure to carry out the non-dominated sorting taking into account the conflict contribution is presented in Algorithm 1.

In extreme cases in which some objectives do not have conflict at all with the other objectives, these objectives do not contribute at all to the structure of the Pareto front, and, therefore, can be removed. Since these objectives do not contribute to the total conflict of the problem, under the new proposed scheme, they will receive no parents. Thus, in these extreme cases, the new scheme can be regarded as a generalization of approaches that remove the least conflicting objectives during the search (see e.g., [10] and [12]).

4.2 Automatic Transition between Phases

In the scheme presented in [15], the number of generations for the partitioning and the integration phases are controlled by a parameter defined by the user. However, the best value for that parameter might differ for different MOPs. In order to free the user from the task of finding the optimal parameter value we designed a mechanism to automate the transition between the partitioning and integration phase automatically. In our approach we employ the progress of the current PF_{approx} in order to decide if the current phase should be changed. That is to say, if the search during the current phase does not have a significant progress, then it is time to change to the other phase. We adopt an approach similar to the convergence detection method proposed in [19]. That approach applies a statistical hypothesis test on a sample of quality indicators values to determine if a MOEA should be stopped. In our case, the convergence detection method will not be used to stop the search, but to switch from one phase to another.

The indicator we propose to measure the progress of the search is based on the additive ϵ-indicator, $I_{\epsilon+}(A, B)$ [16]. This binary indicator is defined as

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{\forall \mathbf{z}^2 \in B \ \exists \mathbf{z}^1 \in A : \mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2\}$$

Algorithm 1. Non-dominated sorting with proportional assignment of parents.

```
procedure SORT&TRUNCATION($\mathcal{R}, |\mathcal{P}|, \Psi$)
    $\mathcal{P}^* \leftarrow \emptyset$
    for $i \leftarrow 1$ until $|\Psi|$ do
        $\mathcal{F}^{\psi_i} \leftarrow$ NONDOMINATEDSORT($\mathcal{R}, \psi_i$)    ▷ Sort using only the objectives in $\psi_i$
        CROWDING($\mathcal{F}^{\psi_i}, \psi_i$)
        $N_P \leftarrow |\mathcal{P}| \times$ conflict contribution of subspace $\psi_i$
        $\mathcal{P}^{\psi_i} \leftarrow$ TRUNCATION($\mathcal{F}^{\psi_i}, N_P$)    ▷ Select the best $N_P$ solutions wrt $\psi_i$
        $\mathcal{P}^* \leftarrow \mathcal{P}^* \cup \mathcal{P}^{\psi_i}$
    return $\mathcal{P}^*$    ▷ $|\mathcal{P}^*| = |\mathcal{P}|$
```

for two nondominated sets A and B, where $\mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2$ iff $\forall i : z_i^1 \leq \epsilon + z_i^2$, for a given ϵ. The proposed indicator is intended to measure the improvement of the current approximation set PF_t with respect to the previous one, PF_{t-1}. We define this improvement as

$$\mathcal{D}_\epsilon(PF_t, PF_{t-1}) = |I_\epsilon(PF_t, PF_{t-1}) - I_\epsilon(PF_{t-1}, PF_t)|.$$

If $\mathcal{D}_\epsilon(PF_t, PF_{t-1})$ is close to zero, that implies that there is no significant improvement from iteration $t-1$ to t. In order to apply a hypothesis test we need to record the values of this indicator through the optimization process to get a sample of $\mathcal{D}_\epsilon$ values. The detailed process is described next. First, we adopt a hypothesis test to detect if the improvement of the search in terms of $\mathcal{D}_\epsilon$ is below a certain threshold. This test is formulated as follows:

$$H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0 \qquad \text{vs.} \qquad H_1 : \mu_{\mathcal{D}_\epsilon} < \mu_0$$

where μ_0 is the degree of improvement required. If H_0 is rejected in support of H_1, we can consider that the search has no significant progress, and, therefore, the phase should be changed. Since the type of distribution of the random variable $\mathcal{D}_\epsilon$ is not known, we employ a non-parametric test, namely, the Wilcoxon signed-rank test [20] to check that hypothesis.

In some preliminary experiments we realized that in some cases, the MOEA diverged (specially for more than 5 objectives) making difficult to reach the desired threshold. Therefore, we also employ a hypothesis test on the slope of the regression line obtained from a sample of D_ϵ values. The test for the analysis regression is formulated in the following manner:

$$H_0 : \beta_{\mathcal{D}_\epsilon} = 0 \qquad \text{vs.} \qquad H_1 : \beta_{\mathcal{D}_\epsilon} \neq 0$$

The slope $\beta_{\mathcal{D}_\epsilon}$ represents the linear trend of the $\mathcal{D}_\epsilon$ indicator. A value of $\beta_{\mathcal{D}_\epsilon} = 0$ means that search is not progressing towards PF_{opt} anymore.

In contrast to the test on $H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0$, in order to determine if there is a significant descending linear trend of $\mathcal{D}_\epsilon$, we need to check the point in which the hypothesis $H_0 : \beta_{\mathcal{D}_\epsilon} = 0$ can no longer be rejected. For this test we use a two-tailed t-test [21].

Since the integration phase is only intended to obtain a new approximation of the Pareto front, and the goal of the partitioning phase is to improve the search ability when the MOEA is stuck in a local optima, we use both hypothesis tests during the partitioning phase. In contrast, during the integration phase we only test the linear trend. One advantage of this is that the value of μ_0 can be automatically set for the test $H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0$. That is to say, we will set $\mu_0 = \hat{\mu}_{\mathcal{D}_\epsilon}$, where $\hat{\mu}_{\mathcal{D}_\epsilon}$ is the mean of the sample of $\mathcal{D}_\epsilon$ taken in the previous integration phase. During the partitioning phase we proceed in the following manner. The test $H_0 : \beta_{\mathcal{D}_\epsilon} = 0$ is checked first, and if the null hypothesis H_0 is rejected, then the partitioning phase is maintained. If that hypothesis is not rejected, then we test $H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0$. The partitioning phase is changed only if H_0 is rejected. Written as a pseudocode, we get the following:

```
If $H_0 : \beta_{\mathcal{D}_\epsilon} = 0$ is not rejected (i.e., search has stalled) and
  $H_1 : \mu_{\mathcal{D}_\epsilon} < \mu_0$ is supported (i.e., no significant improvement) then
    Switch to the integration phase.
Otherwise
    Stay in the partitioning phase.
```

In a similar way to [19], a sample of $\mathcal{D}_\epsilon$ is taken from the last S_N observations of the indicator values. At the beginning of each phase the first test is carried out until S_N observations have been collected. After that, the test is checked during each subsequent generation. Additionally, in the partitioning phase, we manage the progress of each subspace independently to reallocate resources when the search in some of the subspaces has no significant improvement. Therefore, for each subspace in a partition, a sample of the last S_N observation of the D_ϵ indicator is stored. When the search in a certain subspace has stalled, then that subspace is removed from the partition, so that in the next selection of parents, only the remaining subspaces receive parents. This way all the resources are focused on subspaces with a significant progress towards the Pareto front. Fig. 3 shows an example of this procedure applied on a partition with 3 subspaces, i.e., $\Psi = \{\psi_1, \psi_2, \psi_3\}$. Initially, the subspaces receive 30%, 20% and 50% of the parents, respectively. After subspace ψ_2 has been stopped, the proportion of parents assigned to subspaces ψ_1 and ψ_3 is updated as it is shown in the figure. Thus, it is expected to have a convergence speedup in the remaining subspaces. Then, ψ_1 subspace ends, and partition ψ_3 receives all the parents. Finally, ψ_3 is stopped signaling the end of the partitioning phase and the beginning of the integration phase.

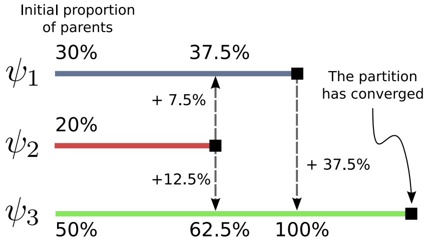


Fig. 3. Dynamic reallocation of the resources granted for each subspace

The pseudocode of the procedure to check convergence is presented in Algorithm 2. In that algorithm α is the significance level used in both hypothesis tests, and *pSlope* and *pWilcoxon* are the p-values returned by the linear regression test and the Wilcoxon test, respectively. Finally, the entire adaptive partitioning scheme is shown in Algorithm 3.

Algorithm 2. Convergence Checking for each partition's subspace

```
procedure CHECKCONVERGENCE($\mathcal{L}, \hat{\mu}_{\mathcal{D}_\epsilon}, \alpha, phase$)
    $convergence_i \leftarrow$ FALSE for all $i \in \{1, \ldots, |\mathcal{L}|\}$
    if $phase =$ INTEGRATION then
        $pSlope \leftarrow$ SLOPETEST($\mathcal{L}_1$)
        if $pSlope > \alpha$ then
            $convergence_1 \leftarrow$ TRUE
    else
        for each $i \in \{1, \ldots, |\Psi|\}$ do
            $pSlope \leftarrow$ SLOPETEST($\mathcal{L}_i$)    ▷ Test using the sample of subspace $\psi_i$
            $pWilcoxon \leftarrow$ WILCOXONTEST($\mathcal{L}_i, \hat{\mu}_{\mathcal{D}_\epsilon}$)
            if $pSlope > \alpha$ and $pWilcoxon \leq \alpha$ then
                $convergence_i \leftarrow$ TRUE
    Return $convergence$
```

Algorithm 3. Adaptive Partitioning MOEA

```
$\Psi \leftarrow \{\{f_1, .., f_M\}\}$    ▷ All the objectives in a single subspace.
$phase \leftarrow$ INTEGRATION
$\mathcal{L}_i \leftarrow \emptyset$ for all $i \in \{1, \ldots, |\Psi|\}$
for $t \leftarrow 1$ until $G_{\max}$ do
    $\mathcal{Q}_t \leftarrow$ NEWPOP($\mathcal{P}_t$)    ▷ selection, crossover, mutation.
    $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$
    $\mathcal{P}_{t+1} \leftarrow$ SORT&TRUNCATION($\mathcal{R}_t, |\mathcal{P}_t|, \Psi$)    ▷ Using current $\Psi$.
    for each $\psi_i \in \Psi$ do
        $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \mathcal{D}_\epsilon(\mathcal{P}_{t+1}, \mathcal{P}_t, \psi_i)$
    if $|\mathcal{L}_1| \geq S_N$ then
        CHECKCONVERGENCE($\mathcal{L}, \hat{\mu}_{\mathcal{D}_\epsilon}, \alpha, phase$)
        if all the subspaces in $\Psi$ have converged then
            if $phase =$ INTEGRATION then
                Compute new $\hat{\mu}_{\mathcal{D}_\epsilon}$ from $\mathcal{L}_1$
                $\Psi \leftarrow$ CREATEPARTITION($\mathcal{P}_{t+1}, \Phi, N_S$)
                $phase \leftarrow$ PARTITIONING
            else
                $\Psi \leftarrow \{\{f_1, .., f_M\}\}$
                $phase \leftarrow$ INTEGRATION
            $\mathcal{L}_i \leftarrow \emptyset$ for all $i \in \{1, \ldots, |\Psi|\}$
        else if at least one subspace have converged then
            Update conflict contribution in the current partition $\Psi$.
```

5 Experimental Results

5.1 Algorithms, Metrics and Parameter Settings

In order to discover the advantages and disadvantages of the new adaptive partitioning scheme we compare 4 versions of the NSGA-II, i.e., the original version and three other versions using different partitioning strategies: random, conflict and adaptive-conflict. In all the algorithms we use a population of 200 individuals running during 200 generations. The results presented are the average over 30 runs of each NSGA-II variant. In the conflict-based strategy, the search is divided into 10 stages, and the values for G_Φ (num. of generations for the integration phase) and G_Ψ (num. of generations for the partitioning phase) represent the 30% and 70% of the generations of each stage, respectively. For the adaptive-conflict strategy we use a significance level $\alpha = 0.05$. The sample size, S_N, for $\mathcal{D}_\epsilon$ was experimentally defined as 10.

In order to show how both conflict-based strategies work, we will use a test problem in which the conflicting objectives can be defined *a priori* by the user. Namely, the problem DTLZ5(I, M) [11], where M is the total number of objectives, and I is the number of objectives in conflict. Additionally, we employ the 0/1 Knapsack with 300 items since the conflict relation among its objectives is not known *a priori*. Unless specified otherwise, in our experiments we use from 4 to 15 objectives in each test problem. For 4-9 objectives we use 2 subspaces, and for 10-15 objectives, we use 3 subspaces. In order to assess convergence we adopt generational distance (GD). For DTZL5(I, M) we use the exact generational distance, namely $GD = \frac{1}{m}\sum_{\mathbf{z}\in PF_{\text{approx}}}\sum_{j=1}^{M}(z_j)^2 - 1$, where $m = |PF_{\text{approx}}|$. Additionally, to directly compare the convergence of the MOEAs, we utilize the additive ϵ-indicator [16]. Finally, to assess both convergence and diversity, we adopt the hypervolume indicator. For DTLZ5(I, M) the reference point was $\mathbf{z}^{\text{ref}} = 1.5^M$. For the Knapsack problem, the reference point was formed using the worst value in each objective of all the PF_{approx} generated by all the algorithms.

5.2 Application of the New Adaptive Scheme

Fig. 4 shows the progress of $\mathcal{D}_\epsilon$ in the Knapsack problem for the partitions of the integration and the partitioning phases. As can be seen, in the last two partitioning phases, the least conflicting subspace (ψ_1) converges first, whereas the most conflicting subspace (ψ_3) converges at the end. This is somewhat expected since the most conflicting subspace has a larger region of the Pareto front to cover.

The assignment of parents for a 3-subspace partition is illustrated in Figure 5. We can observe that the most conflicting subspace receives about 50% of the parents at the beginning of each partitioning phase. Additionally, we can see that after the first subspace converges, the remaining subspaces converge faster than the most conflicting subspace. This can be explained because the remaining subspaces receive more parents to explore their corresponding objective subspace.

5.3 DTLZ5(I, M): Conflict Known *a priori*

With respect to DTLZ5(I, M), the experiments show that the adaptive-conflict strategy only marginally outperforms to the conflict strategy in terms of both

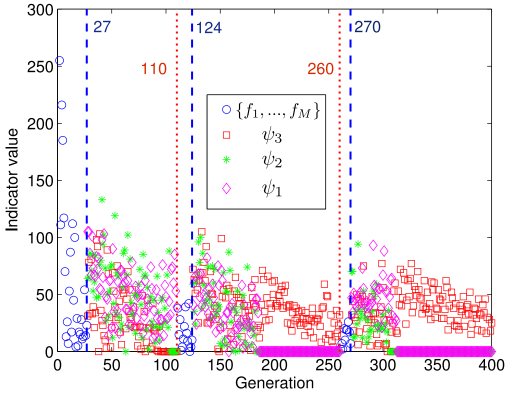


Fig. 4. Example of the convergence of the partitions $\{\{f_1, \ldots, f_M\}\}$ (integration phase) and $\{\psi_1, \psi_2, \psi_3\}$ (partitioning phase) through the search process. Dashed lines mark the end of the integration phase, while dotted lines that of the partitioning phase.

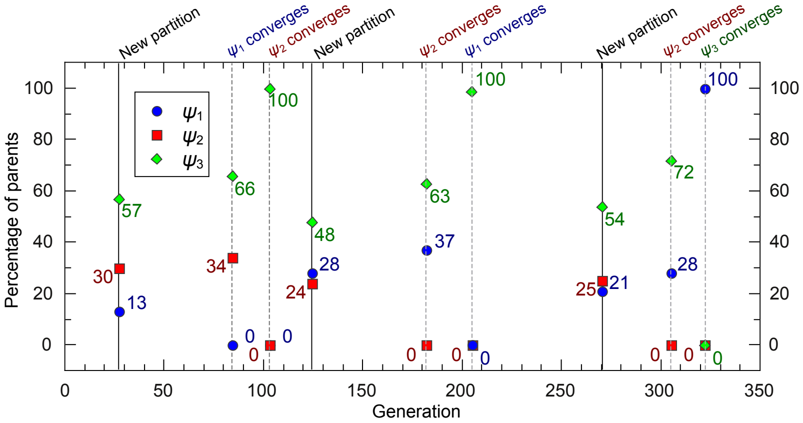


Fig. 5. Example of the proportional assignment of the parents. Solid lines indicate the generation at which a new partition is created, whereas dashed lines indicate the generation at which some subspace has converged. Next to the symbol of each subspace we indicate the number of parents assigned to the subspace.

generational distance and hypervolume (Figs. 6 and 7, respectively). One interesting observation with respect to the hypervolume results is that for less than 5 objectives, NSGA-II performs better than any of the partitioning strategies. This implies that for those number of objectives, NSGA-II still achieves a good performance, and therefore there is no need for partitioning the objective space. Although, of course, the number of objectives in which NSGA-II performs well might depend on the given MOP. Regarding the ϵ-indicator (Fig. 8) the

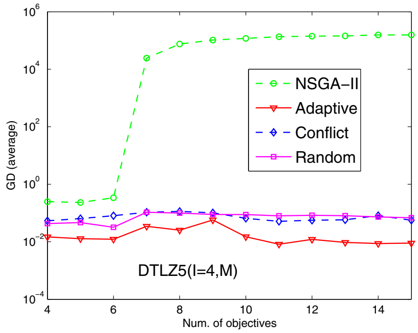


Fig. 6. GD achieved by NSGA-II and the three variants of the partitioning scheme in DTLZ5(I, M) when the number of objectives is varied from 4 to 15 objectives

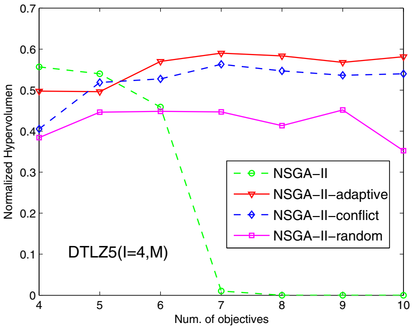


Fig. 7. Hypervolume obtained by NSGA-II and the three variants of the partitioning scheme using the problem DTLZ5(I, M)

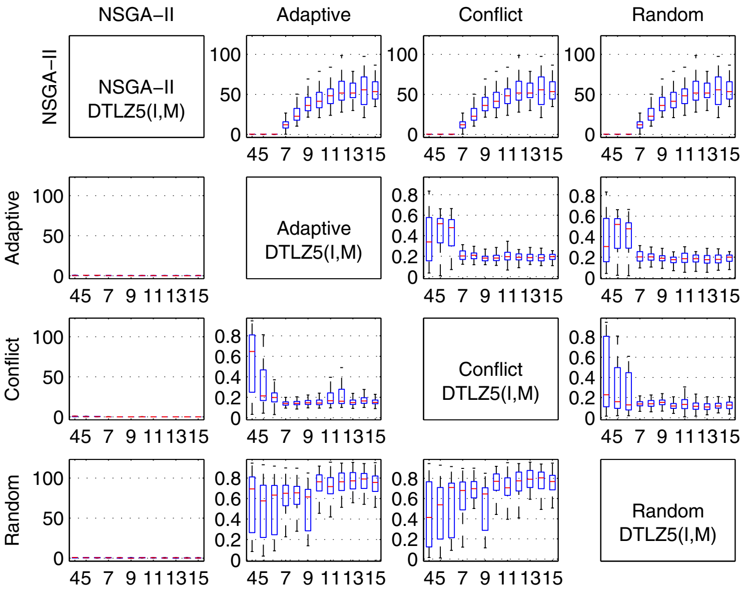


Fig. 8. Comparison of NSGA-II and the variants of the partitioning scheme with respect to I_ϵ using the problem DTLZ5(I, M)

adaptive-conflict strategy clearly outperforms NSGA-II. $I_{\epsilon+}(A, B)$ is the subplot located in row A and column B of the matrix. Nonetheless, the performance of both conflict-based strategies are very similar. From this experiment we can conclude that the proportional assignment of parents accelerates the convergence of the MOEA. Additionally, the automatic transition between phases does not seem to affect the search ability of NSGA-II.

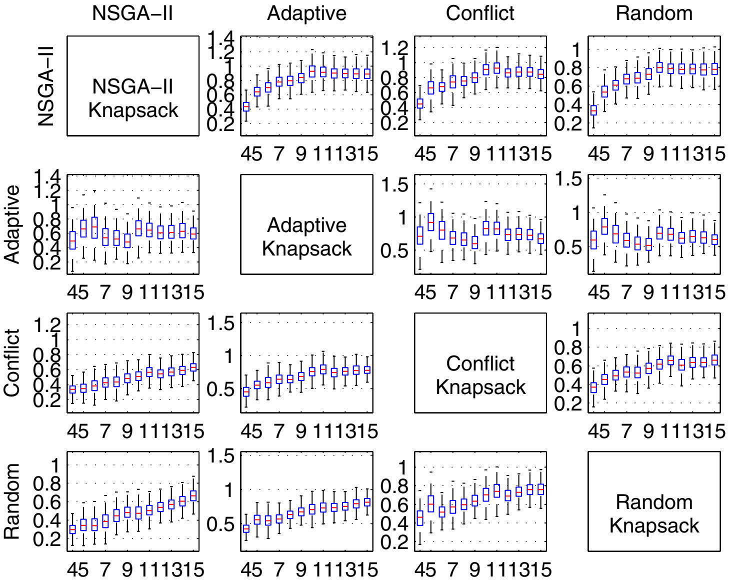


Fig. 9. Comparison of NSGA-II and the variants of the partitioning scheme with respect to I_ϵ using the Knapsack problem

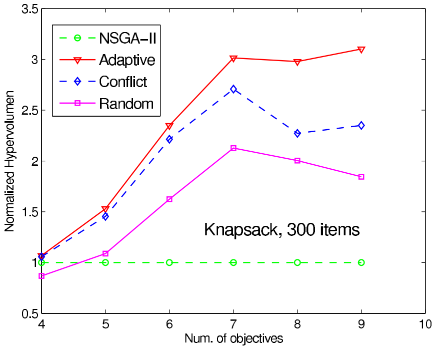


Fig. 10. Hypervolume obtained by NSGA-II and the three variants of the partitioning scheme using the Knapsack problem

5.4 Knapsack Problem: Unknown Conflict *a priori*

One important difference between DTLZ5(I, M) and the Knapsack problem is that none of the objectives of the Knapsack problem is totally redundant, i.e., there is conflict between every pair of objectives. However, as it was shown in [15], the objectives have different conflict degrees among them. Analyzing the boxplots of Fig. 9 we can realize that for a large number of objectives (e.g., 12-15) the adaptive-conflict strategy outperforms the original conflict strategy in terms of the ϵ-indicator 10. This could mean that as the problems get harder, the proportional assignment of parents is more useful. The adaptive-conflict strategy clearly outperforms the other variants of NSGA-II with respect to the hypervolume. This suggests that the adaptive-conflict strategy also contributes to get a better distribution of the solutions.

6 Conclusions and Future Work

In this paper we have introduced some improvements on a previously proposed scheme intended for partitioning the objective space. The main goal of these improvements is to automate the setting of some parameters of the original scheme, namely: the assignment the individuals to explore each subspace, and the transition between each phase of the scheme.

According to the experimental comparison, the new adaptive-conflict partitioning scheme performs at least as well as the two previously defined schemes. Furthermore, in some cases, the adaptive scheme outperformed those schemes. This means that, besides freeing the user from setting the above mentioned parameters, the adaptive scheme improves the search ability of the MOEA.

As a future work, it would be interesting to study the correlation of the transitions between phases and the number of objectives. Additionally, we plan to use the conflict information to determine automatically the size of the subspaces of the created partitions.

Acknowledgements. The first author acknowledges support from CONACyT to pursue graduate studies in Computer Science at CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 103570.

References

1. Coello Coello, C.A., Lamont, G.B.: An Introduction to Multi-Objective Evolutionary Algorithms and Their Applications. In: Coello Coello, C.A., Lamont, G.B. (eds.) Applications of Multi-Objective Evolutionary Algorithms, pp. 1–28. World Scientific, Singapore (2004)
2. Purshouse, R.C., Fleming, P.J.: On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Transactions on Evolutionary Algorithms 11(6), 770–784 (2007)
3. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007)
4. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: CEC 2008, pp. 2424–2431. IEEE Service Center, Hong Kong (2008)
5. Teytaud, O.: On the hardness of offline multi-objective optimization. Evolutionary Computation 15(4), 475–491 (2007)
6. Knowles, J., Corne, D.: Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 757–771. Springer, Heidelberg (2007)

7. Farina, M., Amato, P.: On the Optimal Solution Definition for Many-criteria Optimization Problems. In: Proceedings of the NAFIPS-FLINT International Conference 2002, pp. 233–238. IEEE Service Center, Piscataway (2002)
8. Sato, H., Aguirre, H.E., Tanaka, K.: Controlling Dominance Area of Solutions and its Impact on the Performance of MOEAs. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 5–20. Springer, Heidelberg (2007)
9. Brockhoff, D., Zitzler, E.: Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In: CEC 2007, pp. 2086–2093. IEEE Press, Singapore (2007)
10. López Jaimes, A., Coello Coello, C.A., Urías Barrientos, J.E.: Online Objective Reduction to Deal with Many-Objective Problems. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 423–437. Springer, Heidelberg (2009)
11. Deb, K., Saxena, D.K.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: CEC 2006, pp. 3353–3360. IEEE Press, Vancouver (2006)
12. Brockhoff, D., Zitzler, E.: Are all objectives necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. 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. 533–542. Springer, Heidelberg (2006)
13. López Jaimes, A., Coello Coello, C.A., Chakraborty, D.: Objective Reduction Using a Feature Selection Technique. In: 2008 Genetic and Evolutionary Computation Conference (GECCO 2008), pp. 674–680. ACM Press, Atlanta (2008)
14. Aguirre, H.E., Tanaka, K.: Many-Objective Optimization by Space Partitioning and Adaptive ϵ-Ranking on MNK-Landscapes. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 407–422. Springer, Heidelberg (2009)
15. López Jaimes, A., Aguirre, H., Tanaka, K., Coello Coello, C.: Objective space partitioning using conflict information for many-objective optimization. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 657–666. Springer, Heidelberg (2010)
16. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
17. Carlsson, C., Fullér, R.: Multiple criteria decision making: The case for interdependence. Computers and Operations Research 22(3), 251–260 (1995)
18. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, pp. 93–100. Lawrence Erlbaum, Mahwah (1985)
19. Wagner, T., Trautmann, H., Naujoks, B.: OCD: Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based on Statistical Testing. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 198–215. Springer, Heidelberg (2009)
20. Sheskin, D.: Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC (2004)
21. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers, 4th edn. John Wiley & Sons, Chichester (2006)

Effects of the Existence of Highly Correlated Objectives on the Behavior of MOEA/D

Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Hiroyuki Ohyanagi, and Yusuke Nojima

Department of Computer Science and Intelligent Systems, Graduate School of Engineerig,
Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
{hisaoi@,hitotsu@ci,ohyanagi@ci,nojima@}cs.osakafu-u.ac.jp

Abstract. Recently MOEA/D (multi-objective evolutionary algorithm based on decomposition) was proposed as a high-performance EMO (evolutionary multi-objective optimization) algorithm. MOEA/D has high search ability as well as high computational efficiency. Whereas other EMO algorithms usually do not work well on many-objective problems with four or more objectives, MOEA/D can properly handle them. This is because its scalarizing function-based fitness evaluation scheme can generate an appropriate selection pressure toward the Pareto front without severely increasing the computation load. MOEA/D can also search for well-distributed solutions along the Pareto front using a number of weight vectors with different directions in scalarizing functions. Currently MOEA/D seems to be one of the best choices for multi-objective optimization in various application fields. In this paper, we examine its performance on multi-objective problems with highly correlated objectives. Similar objectives to existing ones are added to two-objective test problems in computational experiments. Experimental results on multi-objective knapsack problems show that the inclusion of similar objectives severely degrades the performance of MOEA/D while it has almost no negative effects on NSGA-II and SPEA2. We also visually examine such an undesirable behavior of MOEA/D using many-objective test problems with two decision variables.

Keywords: Evolutionary multi-objective optimization, evolutionary many-objective optimization, similar objectives, correlated objectives, MOEA/D.

1 Introduction

Since Goldberg's suggestion in 1989 [6], Pareto dominance-based fitness evaluation has been the main stream in the evolutionary multi-objective optimization (EMO) community [3], [26]. Pareto dominance is used for fitness evaluation in almost all well-known and frequently-used EMO algorithms such as NSGA-II [4], SPEA [34] and SPEA2 [33]. Whereas Pareto dominance-based EMO algorithms usually work very well on multi-objective problems with two or three objectives, they often show difficulties in the handling of many-objective problems with four or more objectives as pointed out in several studies [7], [10], [16], [23], [24], [35]. This is because almost all individuals in the current population are non-dominated with each other when they are compared using many objectives. As a result, Pareto dominance-based fitness

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 166–181, 2011.

evaluation cannot generate strong selection pressure toward the Pareto front. This means that good solutions close to the Pareto front are not likely to be obtained.

Various approaches have been proposed to improve the search ability of Pareto dominance-based EMO algorithms for many-objective problems [13], [14]. The basic idea of those approaches is to increase the selection pressure toward the Pareto front. The increase in the selection pressure, however, usually leads to the decrease in the diversity of obtained solutions along the Pareto front. Thus simultaneous performance improvement in both the convergence and the diversity is not easy.

The use of other fitness evaluation schemes has also been examined for many-objective problems. One promising approach to many-objective optimization is the use of an indicator function that measures the quality of a solution set [1], [27], [28], [31], [32]. Hypervolume has been frequently used in such an indicator-based evolutionary algorithm (IBEA) where multi-objective problems are handled as single-objective hypervolume maximization problems. One difficulty of this approach is the exponential increase in the computation load for hypervolume calculation with the increase in the number of objectives. Thus some form of approximate hypervolume calculation may be needed when we have six or more objectives. Another promising approach to many-objective problems is the use of scalarizing functions [8], [15], [19], [29]. A number of scalarizing functions with different weight vectors are used to realize various search directions in the objective space. The main advantage of this approach is computational efficiency of scalarizing function calculation.

MOEA/D (multi-objective evolutionary algorithm based on decomposition) is a scalarizing function-based EMO algorithm proposed by Li and Zhang [19], [29]. High search ability of MOEA/D on various test problems including many-objective problems has already been demonstrated in the literature [2], [11], [12], [17], [20], [22], [30]. Its main feature is the decomposition of a multi-objective problem into a number of single-objective problems, which are defined by a scalarizing function with different weight vectors. MOEA/D can be viewed as a kind of cellular algorithm. Each cell has a different weight vector and a single elite solution with respect to its own weight vector. The task of each cell is to perform single-objective optimization of a scalarizing function with its own weight vector. To generate a new solution for each cell, parents are selected from its neighboring cells (i.e., local parent selection). If a better solution is generated by genetic operations, the current solution is replaced with the newly generated one. This solution replacement mechanism is applied to not only the current cell for which a new solution is generated but also its neighboring cells. That is, a good solution has a chance to survive at multiple cells. Such a local solution replacement mechanism together with local parent selection accelerates multi-objective search for better solutions (i.e., accelerates the convergence toward the Pareto front). At the same time, the diversity of solutions is maintained by the use of a number of weight vectors with various directions in MOEA/D.

In this paper, we report some interesting observations on the behavior of NSGA-II, SPEA2 and MOEA/D on multi-objective problems with highly correlated objectives. In computational experiments, we generate similar objectives to existing ones and add them to test problems with two objectives. Experimental results show that the inclusion of similar objectives severely deteriorates the search ability of MOEA/D while it has almost no negative effects on NSGA-II and SPEA2. As a result, MOEA/D does not always outperform NSGA-II and SPEA2 on many-objective problems with highly correlated objectives while it clearly shows better performance on many-objective problems with no strong correlations among objectives.

This paper is organized as follows. First we briefly explain MOEA/D in Section 2. Next we examine its behavior on two types of multi-objective knapsack problems in comparison with NSGA-II and SPEA2 in Section 3. One type has randomly generated objectives, and the other includes highly correlated objectives. Then we visually examine the behavior of MOEA/D using many-objective test problems in the two-dimensional decision space in Section 4. Finally we conclude this paper in Section 5.

2 MOEA/D

Let us consider the following m-objective maximization problem:

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ ...,\ f_m(\mathbf{x}))\,, \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is the m-dimensional objective vector, $f_i(\mathbf{x})$ is the i-th objective to be maximized, and $\mathbf{x}$ is the decision vector.

In MOEA/D [19], [29], a multi-objective problem is decomposed into a number of single-objective problems where each problem is to optimize a scalarizing function with a different weight vector. In this paper, we use the weighted Tchebycheff function since this function works very well on a wide range of multi-objective test problems [9], [11], [12]. Let us denote a weight vector as $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)$. The weighted Tchebycheff function measures the distance from the reference point $\mathbf{z}^*$ to a solution $\mathbf{x}$ in the objective space as follows:

$$g^{TE}(\mathbf{x}\,|\,\boldsymbol{\lambda}, \mathbf{z}^*) = \max_{i=1,2,...,m} \{\lambda_i \cdot |\, z_i^* - f_i(\mathbf{x})\,|\}\,. \tag{2}$$

For multi-objective knapsack problems, we use the following specification of the reference point $\mathbf{z}^*$ in the same manner as in Zhang and Li [29]:

$$z_i^* = 1.1 \cdot \max\{f_i(\mathbf{x})\,|\,\mathbf{x} \in \Omega(t)\},\ i = 1, 2, ..., m\,, \tag{3}$$

where $\Omega(t)$ shows the population at the t-th generation. The reference point $\mathbf{z}^*$ is updated whenever the maximum value of each objective in (3) is updated.

For multi-objective function minimization problems, Zhang and Li [29] specified the reference point $\mathbf{z}^*$ as follows:

$$z_i^* = \min\{f_i(\mathbf{x})\,|\,\mathbf{x} \in \Omega(t)\},\ i = 1, 2, ..., m\,. \tag{4}$$

We use this specification for multi-objective continuous minimization problems.

MOEA/D uses a set of weight vectors placed on a uniform grid. More specifically, it uses all weight vectors satisfying the following two conditions:

$$\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1\,, \tag{5}$$

$$\lambda_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, ..., \frac{H}{H}\right\},\ i = 1, 2, ..., m\,, \tag{6}$$

where H is a positive integer parameter that specifies the granularity or resolution of weight vectors. The number of weight vectors is calculated from this parameter H and the number of objectives m as $N = {}_{H+m-1}C_{m-1}$ [29]. The same weight vector specification was used in a multi-objective cellular algorithm in Murata et al. [21].

Let N be the number of weight vectors. Then a multi-objective optimization problem is decomposed into N single-objective problems in MOEA/D. Each single-objective problem has the same scalarizing function (i.e., the weighted Tchebycheff function in this paper) with a different weight vector. Each weight vector can be viewed as a cell in a cellular algorithm with a grid of size N in the m-dimensional unit cube $[0, 1]^m$. A single individual is assigned to each cell. Thus the population size is the same as the number of weight vectors. In MOEA/D, genetic operations at each cell are locally performed within its neighboring cells as in cellular algorithms. For each cell, a pre-specified number of its nearest cells (e.g., ten cells including the cell itself in our computational experiments) are handled as its neighbors. Neighborhood structures in MOEA/D are defined by the Euclidean distance between weight vectors.

First MOEA/D generates an initial solution at each cell. In our computational experiments, initial solutions are randomly generated. Next an offspring is generated by local selection, crossover and mutation at each cell in an unsynchronized manner. In local selection, two parents are randomly chosen for the current cell from its neighboring cells (including the current cell itself). Local selection leads to the recombination of similar parents in the objective space. The generated offspring is compared with the solution at each of the neighboring cells. The comparison is performed based on the scalarizing function with the weight vector of the compared neighbor. All the inferior solutions are replaced with the newly generated offspring. That is, solution replacement is performed not only at the current cell for which the new offspring is generated but also at each of its neighboring cells. Since each cell has a different weight vector, the diversity of solutions can be maintained whereas a single offspring is compared with multiple neighbors for solution replacement. Local selection, crossover, mutation and local replacement are performed at each cell. These procedures are iterated over all cells until the termination condition is satisfied. In our computational experiments, we do not use any secondary population in MOEA/D.

3 Computational Experiments on Knapsack Problems

We used the same test problem as the two-objective 500-item knapsack problem of Zitzler & Thiele [34] with two constraint conditions. We denote this test problem as the 2-500 problem. The two objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ of the 2-500 problem were generated by randomly assigning an integer in the closed interval [10, 100] to each item as its profit (see [34]). In the same manner, we generated other two objectives $f_3(\mathbf{x})$ and $f_4(\mathbf{x})$. Since all the four objectives were randomly generated, they have no strong correlation with each other. Using these randomly-generated four objectives, we generated a four-objective 500-item knapsack problem with the same two constraint conditions as in the 2-500 problem in [34]. Exactly the same two constraint conditions as in [34] were also used in all the other test problems in this section.

We generated highly correlated objectives from the two objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ of the 2-500 problem in the following manner:

$$f_5(\mathbf{x}) = f_1(\mathbf{x}) + 0.01 f_2(\mathbf{x})\,, \tag{7}$$

$$f_6(\mathbf{x}) = f_1(\mathbf{x}) - 0.01 f_2(\mathbf{x})\,, \tag{8}$$

$$f_7(\mathbf{x}) = f_2(\mathbf{x}) + 0.01 f_1(\mathbf{x})\,, \tag{9}$$

$$f_8(\mathbf{x}) = f_2(\mathbf{x}) - 0.01 f_1(\mathbf{x})\,. \tag{10}$$

It is clear that $f_5(\mathbf{x})$ and $f_6(\mathbf{x})$ are similar to $f_1(\mathbf{x})$ while $f_7(\mathbf{x})$ and $f_8(\mathbf{x})$ are similar to $f_2(\mathbf{x})$. In computational experiments, we used the following four test problems:

1. The 2-500 test problem with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ of Zitzler & Thiele [34],
2. Random four-objective problem with $f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})$ and $f_4(\mathbf{x})$,
3. Correlated four-objective problem with $f_1(\mathbf{x}), f_2(\mathbf{x}), f_5(\mathbf{x})$ and $f_6(\mathbf{x})$,
4. Correlated six-objective problem with $f_1(\mathbf{x}), f_2(\mathbf{x}), f_5(\mathbf{x}), f_6(\mathbf{x}), f_7(\mathbf{x})$ and $f_8(\mathbf{x})$.

In the correlated four-objective problem, $f_1(\mathbf{x}), f_5(\mathbf{x})$ and $f_6(\mathbf{x})$ are similar to each other while they are not similar to $f_2(\mathbf{x})$. In the correlated six-objective problem, $f_2(\mathbf{x}), f_7(\mathbf{x})$ and $f_8(\mathbf{x})$ are also similar to each other.

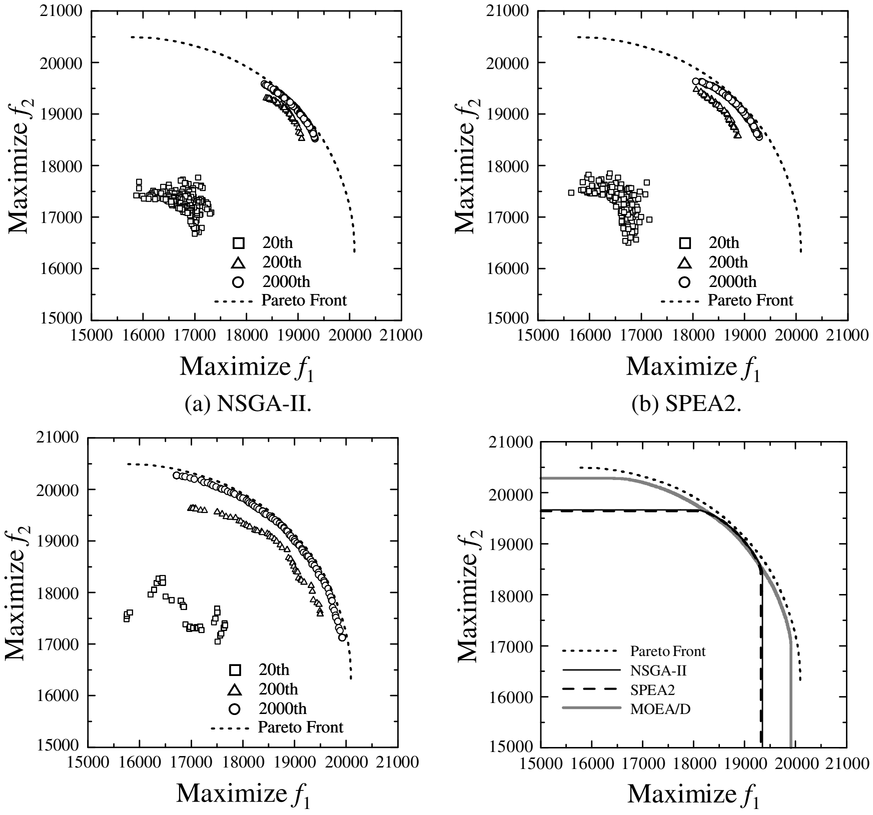
We applied MOEA/D to these four test problems using the following setting:

Population size (which is the same as the number of weight vectors):
200 (two-objective problem), 220 (four-objective), 252 (six-objective),
Parameter H for generating weight vectors:
199 (two-objective problem), 9 (four-objective), 5 (six-objective),
Coding: Binary string of length 500,
Stopping condition: 400,000 solution evaluations,
Crossover probability: 0.8 (Uniform crossover),
Mutation probability: 1/500 (Bit-flip mutation),
Constraint handling: Greedy repair used in Zitzler & Thiele [34],
Neighborhood size T (i.e., the number of neighbors): 10.

Since all the four test problems have the same constraint conditions, the same greedy repair as in the 2-500 problem in [34] was used in all test problems.

We also applied NSGA-II [4] and SPEA2 [33] to the four test problems using the same setting as in MOEA/D except that the population size was always specified as 200 in NSGA-II and SPEA2. Each EMO algorithm was applied to each test problem 100 times. In this section, we report experimental results by NSGA-II, SPEA2 and MOEA/D on each of the above-mentioned four test problems.

Results on the 2-500 Knapsack Problem: Experimental results of a single run of each algorithm are shown in Fig. 1 (a)-(c) where all solutions at the 20th, 200th and 2000th generations are depicted together with the true Pareto front. The 50% attainment surface [5] at the 2000th generation over 100 runs of each algorithm is depicted in Fig. 1 (d). As pointed out in the literature (e.g., see Jaszkiewicz [15]), it is not easy for EMO algorithms to find non-dominated solutions along the entire Pareto front of the 2-500 test problem. NSGA-II and SPEA2 found non-dominated solutions around the center of the Pareto front. Only MOEA/D found non-dominated solutions over almost the entire Pareto front. That is, MOEA/D found better solution sets than NSGA-II and SPEA2 with respect to the diversity of solutions along the Pareto front. With respect to the convergence property toward the Pareto front, the three algorithms have almost the same performance in Fig. 1 (d).



(a) NSGA-II.

(b) SPEA2.

(c) MOEA/D.

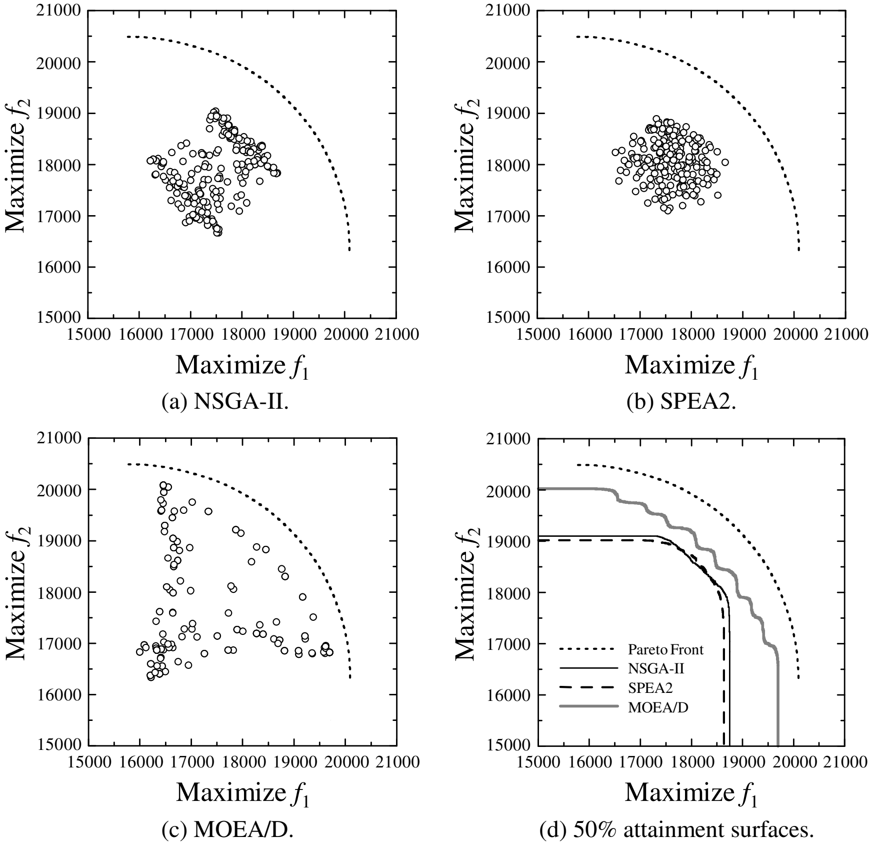
(d) 50% attainment surfaces.

Fig. 1. Experimental results on the original two-objective knapsack problem

Results on Random Four-Objective Knapsack Problem: We calculated the average hypervolume and the standard deviation over 100 runs of each algorithm. The origin (0, 0, 0, 0) of the four-dimensional (4-D) objective space was used as the reference point for the hypervolume calculation. The following results were obtained:

NSGA-II: 1.23×10^{17} (Average), 9.67×10^{14} (Standard Deviation),
SPEA2: 1.19×10^{17} (Average), 9.47×10^{14} (Standard Deviation),
MOEA/D: 1.43×10^{17} (Average), 6.00×10^{14} (Standard Deviation).

The best results were obtained by MOEA/D for the random four-objective problem with respect to the hypervolume measure. In Fig. 2 (a)-(c), we show all solutions at the final generation in a single run of each algorithm in the two-dimensional (2-D) objective space with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. That is, each plot in Fig. 2 shows the projection of the final population of each algorithm in the 4-D objective space onto the 2-D objective space. The 50% attainment surface is depicted using those projections for 100 runs in Fig. 2 (d). For comparison, the Pareto front of the 2-500 problem is also shown in Fig. 2. In Fig. 2 (d), the convergence performance of NSGA-II and SPEA2 was severely degraded by the inclusion of the randomly generated objectives $f_3(\mathbf{x})$ and $f_4(\mathbf{x})$. The convergence performance of MOEA/D was also degraded but less severely than NSGA-II and SPEA2. It is interesting to observe that NSGA-II in Fig. 2 (a) and SPEA2 in Fig. 2 (b) did not have large diversity in the objective space even in the case of the four-objective test problem with the randomly generated objectives.



(a) NSGA-II.

(b) SPEA2.

(c) MOEA/D.

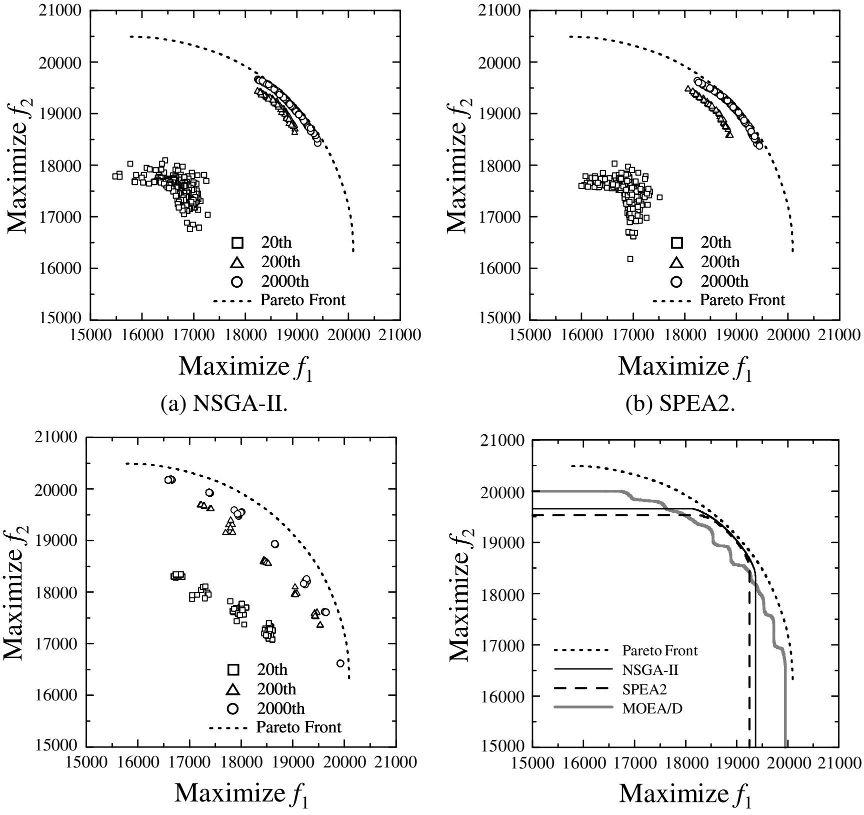
(d) 50% attainment surfaces.

Fig. 2. Experimental results on the random four-objective knapsack problem (Projections of the final population in the 4-D objective space onto the 2-D one)

Results on Correlated Four-Objective Knapsack Problem: This problem has the three highly correlated objectives $f_1(\mathbf{x})$, $f_5(\mathbf{x})$ and $f_6(\mathbf{x})$. As in the previous computational experiments, we calculated the average hypervolume over 100 runs:

NSGA-II: 1.42×10^{17} (Average), 1.59×10^{15} (Standard Deviation),
SPEA2: 1.41×10^{17} (Average), 1.32×10^{15} (Standard Deviation),
MOEA/D: 1.55×10^{17} (Average), 7.46×10^{14} (Standard Deviation).

The best average result was obtained by MOEA/D. In the same manner as Fig. 2, we show experimental results on the correlated four-objective problem in Fig. 3. From the comparison between Fig. 1 and Fig. 3, we can see that the inclusion of $f_5(\mathbf{x})$ and $f_6(\mathbf{x})$ had almost no negative effects on the performance of NSGA-II and SPEA2. The performance of MOEA/D, however, was clearly degraded by their inclusion. We can also see that many solutions in Fig. 3 (c) are overlapping, which leads to the wavy 50% attainment surface by MOEA/D in Fig. 3 (d). In spite of the performance deterioration, the largest average hypervolume was still obtained by MOEA/D.



(a) NSGA-II.

(b) SPEA2.

(c) MOEA/D.

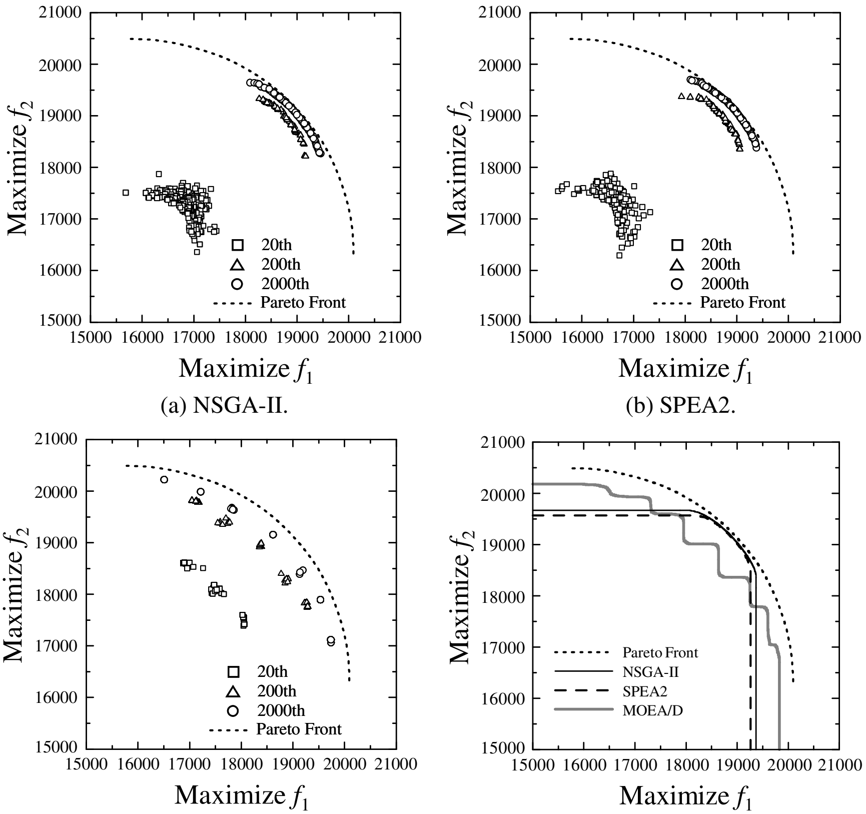
(d) 50% attainment surfaces.

Fig. 3. Experimental results on the correlated four-objective knapsack problem (Projections from the 4-D objective space to the 2-D one). The number of generations of MOEA/D with population size 220 was converted to the equivalent one in the case of population size 200.

Results on Correlated Six-Objective Knapsack Problem: This problem has the two sets of the three highly correlated objectives: $\{f_1(\mathbf{x}), f_5(\mathbf{x}), f_6(\mathbf{x})\}$ and $\{f_2(\mathbf{x}), f_7(\mathbf{x}), f_8(\mathbf{x})\}$. We calculated the average hypervolume over 100 runs:

NSGA-II: 5.46×10^{25} (Average), 5.69×10^{23} (Standard Deviation),
SPEA2: 5.40×10^{25} (Average), 4.59×10^{23} (Standard Deviation),
MOEA/D: 5.87×10^{25} (Average), 4.72×10^{23} (Standard Deviation).

As in the other three test problems, the best results were obtained by MOEA/D with respect to the hypervolume measure. In the same manner as Fig. 2 and Fig. 3, we show experimental results in the two-dimensional (2-D) objective space in Fig. 4. From the comparison between Fig. 1 and Fig. 4, we can see that the inclusion of the four correlated objectives had almost no negative effects on the performance of NSGA-II and SPEA2. The performance of MOEA/D, however, was clearly degraded by their inclusion. That is, the convergence performance was degraded and the number of obtained solutions was decreased (see Fig. 1 (c), Fig. 3 (c) and Fig. 4 (c)).



(a) NSGA-II.

(b) SPEA2.

(c) MOEA/D.

(d) 50% attainment surfaces.

Fig. 4. Experimental results on the correlated six-objective knapsack problem (Projections from the 6-D objective space to the 2-D one). The number of generations of MOEA/D with population size 252 was converted to the equivalent one in the case of population size 200.

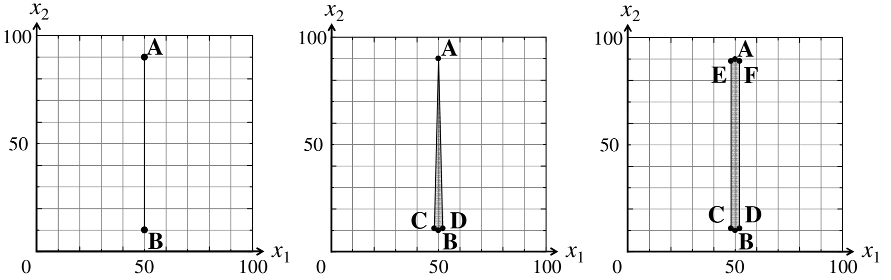
As in Figs. 2-4, we projected the solution set obtained by each run onto the 2-D objective space with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. Then we calculated its hypervolume in the 2-D objective space using the origin (0, 0) as the reference point. Table 1 summarizes the average result over 100 runs of each algorithm on each test problem. When we included the two randomly generated objectives into the 2-500 knapsack problem, the performance of NSGA-II and SPEA2 was severely degraded (see the row labeled as "Random 4-Obj." in Table 1). However, the inclusion of the two and four correlated objectives did not degrade their performance at all. MOEA/D shows a totally different behavior from the others. The performance of MOEA/D was clearly degraded by the inclusion of the correlated objectives as well as the randomly generated objectives. In spite of the performance deterioration, the best results were obtained by MOEA/D for all the four problems in Table 1.

Table 1. Average hypervolume and standard deviation in the original two-objective space

Problem	NSGA-II		SPEA2		MOEA/D	
	Average	Stand. Dev.	Average	Stand. Dev.	Average	Stand. Dev.
Original 2-Obj.	3.800E+08	1.764E+06	3.790E+08	1.390E+06	4.005E+08	9.291E+05
Random 4-Obj.	3.577E+08	2.184E+06	3.537E+08	1.862E+06	3.902E+08	1.435E+06
Correlated 4-Obj.	3.800E+08	1.618E+06	3.790E+08	1.255E+06	3.949E+08	1.491E+06
Correlated 6-Obj.	3.804E+08	1.483E+06	3.782E+08	5.481E+06	3.947E+08	1.194E+06

4 Computational Experiments on Two-Dimensional Problems

In this section, we visually examine the behavior of EMO algorithms using test problems with only two decision variables. In our test problems, the distance to each of the given points in the two-dimensional decision space [0, 100] × [0, 100] is minimized. In Fig. 5, we show our three test problems used in this section.



(a) Two-objective problem. (b) Four-objective problem. (c) Six-objective problem.

Fig. 5. Three test problems used in Section 4

For example, let us assume that four points A, B, C and D are given as in Fig. 5 (b). In this case, our four-objective test problem is written as follows:

$$\text{Minimize } \textit{distance}(\mathrm{A}, \mathbf{x}), \textit{distance}(\mathrm{B}, \mathbf{x}), \textit{distance}(\mathrm{C}, \mathbf{x}) \text{ and } \textit{distance}(\mathrm{D}, \mathbf{x}), \tag{11}$$

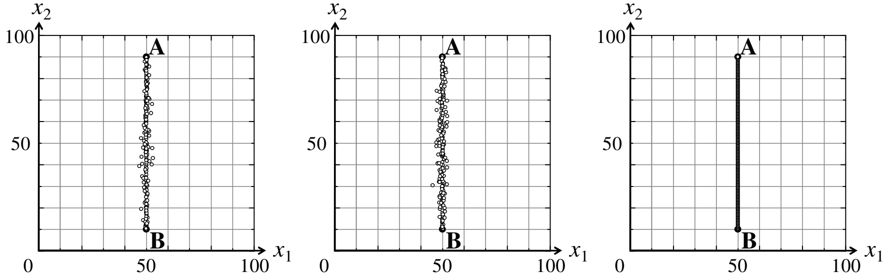
where *distance*(A, **x**) shows the Euclidean distance between the point A and a two-dimensional decision vector **x** in the decision space [0, 100] × [0, 100].

As shown in this formulation, the number of objectives is the same as the number of the given points. Thus we can generate various test problems with an arbitrary number of objectives. As in Fig. 5, we can also generate highly correlated objectives using closely located points. Regular polygons were used to generate this type of test problems in [18], [25]. Multiple polygons were used to generate test problems with multiple equivalent Pareto regions and/or disjoint Pareto regions in [9].

We applied NSGA-II, SPEA2 and MOEA/D to our three test problems in Fig. 5 using the following setting:

Population size in NSGA-II and SPEA2: 200
Population size in MOEA/D: 200 (2-objective), 220 (4-objective), 252 (6-objective),
Parameter H for generating weight vectors in MOEA/D:
199 (2-objective), 9 (4-objective), 5 (6-objective),
Coding: Real number string of length 2,
Stopping condition: 400,000 solution evaluations,
Crossover probability: 1.0 (SBX with $\eta_c = 15$),
Mutation probability: 0.5 (Polynomial mutation with $\eta_m = 20$),
Neighborhood size T (i.e., the number of neighbors): 10.

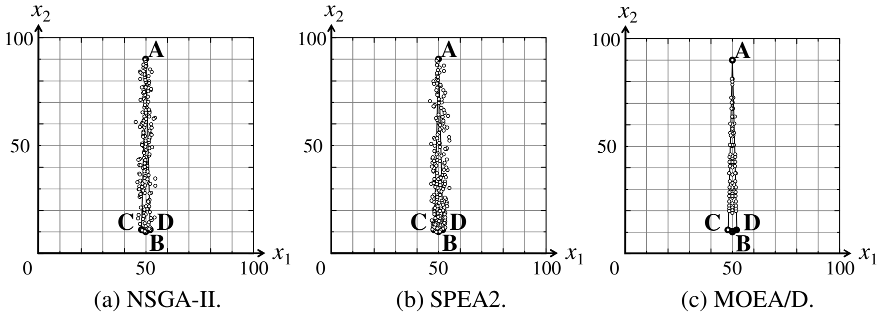
Experimental results of a single run of each algorithm on the two-objective test problem in Fig. 5 (a) are shown in Fig. 6. All solutions at the final generation are shown in each plot in Fig. 6 where Pareto optimal solutions are points on the line between the points A and B. In Fig. 6 (a) and Fig. 6 (b), the final populations included many sub-optimal solutions that are not on the line between the points A and B. Much better results with respect to the convergence to the Pareto front were obtained by MOEA/D in Fig. 6 (c) where all solutions are on the line between the points A and B.



(a) NSGA-II. (b) SPEA2. (c) MOEA/D.

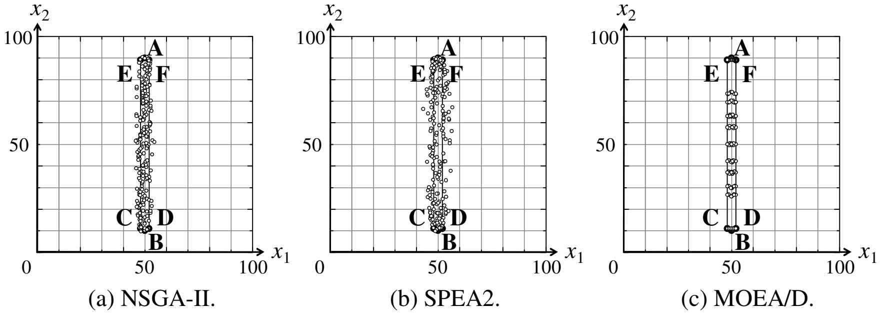
Fig. 6. Experimental results on the two-objective test problem in Fig. 5 (a)

Fig. 7 shows experimental results on the four-objective test problem in Fig. 5 (b) where Pareto optimal solutions are points inside the four points A, B, C and D. As in Fig. 6, the best results with respect to the convergence were obtained by MOEA/D in Fig. 7. However, we can observe some regions with no solutions in Fig. 7 (c). That is, obtained solutions in Fig. 7 (c) are not uniformly distributed.



(a) NSGA-II. (b) SPEA2. (c) MOEA/D.

Fig. 7. Experimental results on the four-objective test problem in Fig. 5 (b)



(a) NSGA-II. (b) SPEA2. (c) MOEA/D.

Fig. 8. Experimental results on the six-objective test problem in Fig. 5 (c)

Fig. 8 shows experimental results on the six-objective test problem in Fig. 5 (c) where Pareto optimal solutions are points inside the six points A, B, C, D, E and F.

As in Fig. 6 and Fig. 7, the best results with respect to the convergence were obtained by MOEA/D in Fig. 8. However, we can observe that obtained solutions in Fig. 8 (c) by MOEA/D are not uniformly distributed.

We calculated the average hypervolume over 100 runs. The reference point for hypervolume calculation was specified as 1.1 × (the maximum objective value for each objective among Pareto optimal solutions of each problem), which is 1.1 × (the distance from each point to its farthest point). Experimental results are summarized in Table 2. For the two-objective problem, the best results were obtained by MOEA/D in Table 2, which is consistent with Fig. 6. The performance of MOEA/D, however, was the worst in Table 2 for the four-objective and six-objective problems. This is due to the existence of regions with no solutions as shown in Fig. 7 (c) and Fig. 8 (c).

Table 2. Average hypervolume and standard deviation for each test problem

Problem	NSGA-II		SPEA2		MOEA/D	
	Average	Stand. Dev.	Average	Stand. Dev.	Average	Stand. Dev.
Two-Objective	4.521E+03	7.409E-01	4.518E+03	1.616E+00	4.528E+03	3.801E-03
Four-Objective	1.992E+07	1.720E+04	1.987E+07	2.694E+04	1.912E+07	6.626E+02
Six-Objective	3.761E+10	4.322E+07	3.736E+10	6.520E+07	3.227E+10	6.724E+06

We also calculated the average hypervolume in the 2-D objective space with the two objectives defined by the two points A and B. That is, solution sets obtained for the four-objective and six-objective problems were projected onto the two-dimensional (2-D) objective space as in Table 1 in the previous section. Then the average hypervolume were calculated. Experimental results are summarized in Table 3. When the three algorithms were applied to the two-objective problem, the best average value 4.528×10^3 was obtained by MOEA/D. However, this average value was decreased to 4.216×10^3 by 6.89% when MOEA/D was applied to the six-objective problem. In the case of NSGA-II, the decrease in the average hypervolume value was only 0.35% from 4.521×10^3 to 4.505×10^3. The decrease in the case of SPEA2 was also small (i.e., 0.58%). That is, the inclusion of the highly correlated objectives severely degraded the performance of MOEA/D whereas it had almost no negative effects on the other two algorithms.

In Fig. 7 (c) and Fig. 8 (c), solution sets with strange distributions were obtained by MOEA/D for the four-objective and six-objective problems. Such an undesirable behavior disappeared when we decreased the correlation among the similar objectives (i.e., when we increased the distance among the closely located points as shown in Fig. 9). From Fig. 9, we can see that the decrease in the correlation among the objectives leads to better distributions along the line between the two points A and B.

Table 3. Average hypervolume and standard deviation in the 2-D objective space

Problem	NSGA-II		SPEA2		MOEA/D	
	Average	Stand. Dev.	Average	Stand. Dev.	Average	Stand. Dev.
Two-Objective	4.521E+03	7.409E-01	4.518E+03	1.616E+00	4.528E+03	3.801E-03
Four-Objective	4.504E+03	2.894E+00	4.485E+03	4.198E+00	4.412E+03	1.380E-01
Six-Objective	4.505E+03	2.013E+00	4.492E+03	3.283E+00	4.216E+03	1.216E-01

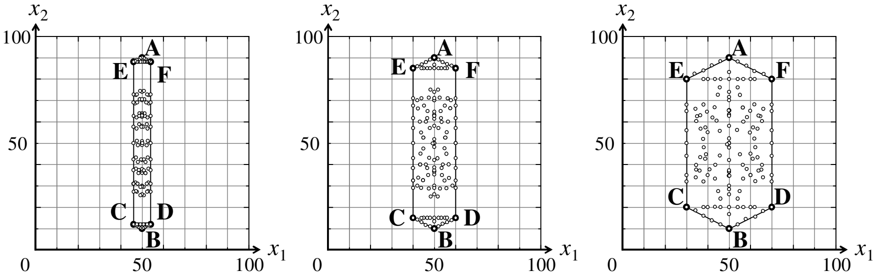


Fig. 9. Experimental results of MOEA/D on other six-objective test problems

5 Conclusions

We demonstrated that highly correlated objectives severely degraded the performance of MOEA/D whereas they had almost no negative effects on the performance of NSGA-II and SPEA2. The reason for the performance deterioration of MOEA/D may be the use of a uniform grid of weight vectors independent of the correlation among objectives. When an m-objective problem has highly correlated objectives, its Pareto front in the objective space has a totally different shape from the uniform grid of weight vectors in the m-dimensional weight vector space $[0, 1]^m$. This leads to a strange distribution of obtained solutions by MOEA/D. Our experimental results clearly suggest the necessity of the adjustment of weight vectors according to the correlation among objectives, which is left as a future research issue. Further discussions on the behavior of each algorithm are also left as a future research issue.

References

1. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective Selection based on Dominated Hypervolume. European Journal of Operational Research 181, 1653–1669 (2007)
2. Chang, P.C., Chen, S.H., Zhang, Q., Lin, J.L.: MOEA/D for Flowshop Scheduling Problems. In: Proc. of 2008 IEEE Congress on Evolutionary Computation, pp. 1433–1438 (2008)
3. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6, 182–197 (2002)
5. Fonseca, C.M., Fleming, P.J.: On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 584–593. Springer, Heidelberg (1996)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
7. Hughes, E.J.: Evolutionary Many-Objective Optimization: Many Once or One Many? In: Proc. of 2005 IEEE Congress on Evolutionary Computation, pp. 222–227 (2005)
8. Hughes, E.J.: MSOPS-II: A General-Purpose Many-Objective Optimizer. In: Proc. of 2007 IEEE Congress on Evolutionary Computation, pp. 3944–3951 (2007)
9. Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., Nojima, Y.: Many-Objective Test Problems to Visually Examine the Behavior of Multiobjective Evolution in a Decision Space. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 91–100. Springer, Heidelberg (2010)
10. Ishibuchi, H., Nojima, Y., Doi, T.: Comparison between Single-objective and Multi-objective Genetic Algorithms: Performance Comparison and Performance Measures. In: Proc. of 2006 IEEE Congress on Evolutionary Computation, pp. 3959–3966 (2006)
11. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Evolutionary Many-objective Optimization by NSGA-II and MOEA/D with Large Populations. In: Proc. of 2009 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1820–1825 (2009)
12. Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y.: Simultaneous Use of Different Scalarizing Functions in MOEA/D. In: Proc. of 2010 Genetic and Evolutionary Computation Conference, pp. 519–526 (2010)
13. Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., Nojima, Y.: Effectiveness of Scalability Improvement Attempts on the Performance of NSGA-II for Many-Objective Problems. In: Proc. of 2008 Genetic and Evolutionary Computation Conference, pp. 649–656 (2008)

14. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary Many-Objective Optimization: A Short Review. In: Proc. of 2008 IEEE Congress on Evolutionary Computation, pp. 2424–2431 (2008)
15. Jaszkiewicz, A.: On the Computational Efficiency of Multiple Objective Metaheuristics: The Knapsack Problem Case Study. European Journal of Operational Research 158, 418–433 (2004)
16. Khara, V., Yao, X., Deb, K.: Performance Scaling of Multi-objective Evolutionary Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 367–390. Springer, Heidelberg (2003)
17. Konstantinidis, A., Yang, K., Zhang, Q.F., Zeinalipour-Yazti, D.: A Multi-objective Evolutionary Algorithm for the Deployment and Power Assignment Problem in Wireless Sensor Networks. Computer Networks 54, 960–976 (2010)
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. Li, H., Zhang, Q.F.: A Multiobjective Differential Evolution Based on Decomposition for Multiobjective Optimization with Variable Linkages. 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. 583–592. Springer, Heidelberg (2006)
20. Li, H., Zhang, Q.: Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Trans. on Evolutionary Computation 13, 284–302 (2009)
21. Murata, T., Ishibuchi, H., Gen, M.: Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 82–95. Springer, Heidelberg (2001)
22. Peng, W., Zhang, Q., Li, H.: Comparison between MOEA/D and NSGA-II on the Multi-objective Travelling Salesman Problem. In: Goh, C.K., Ong, Y.S., Tan, K.C. (eds.) Multi-Objective Memetic Algorithms, pp. 309–324. Springer, Heidelberg (2009)
23. Purshouse, R.C., Fleming, P.J.: Evolutionary Many-Objective Optimization: An Exploratory Analysis. In: Proc. of 2003 IEEE Congress on Evolutionary Computation, pp. 2066–2073 (2003)
24. Purshouse, R.C., Fleming, P.J.: On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Trans. on Evolutionary Computation 11, 770–784 (2007)
25. Singh, H., Isaacs, A., Ray, T., Smith, W.: A Study on the Performance of Substitute Distance Based Approaches for Evolutionary Many Objective Optimization. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 401–410. Springer, Heidelberg (2008)
26. Tan, K.C., Khor, E.F., Lee, T.H.: Multiobjective Evolutionary Algorithms and Applications. Springer, Berlin (2005)
27. Ulrich, T., Bader, J., Zitzler, E.: Integrating Decision Space Diversity into Hypervolume-Based Multiobjective Search. In: Proc. of 2010 Genetic and Evolutionary Computation Conference, pp. 455–462 (2010)
28. Wagner, T., Beume, N., Naujoks, B.: Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007)
29. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Trans. on Evolutionary Computation 11, 712–731 (2007)

30. Zhang, Q.F., Liu, W.D., Tsang, E., Virginas, B.: Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model. IEEE Trans. on Evolutionary Computation 14, 456–474 (2010)
31. Zitzler, E., Brockhoff, D., Thiele, L.: The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 862–876. Springer, Heidelberg (2007)
32. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
33. Zitzler, E., Laumanns, M., Thiele L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, ETH, Zurich (2001)
34. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. on Evolutionary Computation 3, 257–271 (1999)
35. Zou, X., Chen, Y., Liu, M., Kang, L.: A New Evolutionary Algorithm for Solving Many-Objective Optimization Problems. IEEE Trans. on Systems, Man, and Cybernetics: Part B - Cybernetics 38, 1402–1412 (2008)

Improved Random One-Bit Climbers with Adaptive ε-Ranking and Tabu Moves for Many-Objective Optimization

Joseph M. Pasia[1,2], Hernán Aguirre[1], and Kiyoshi Tanaka[1]

[1] Faculty of Engineering, Shinshu University
4-17-1 Wakasato, Nagano 380-8553, Japan
{ahernan,ktanaka}@shinshu-u.ac.jp
[2] Institute of Mathematics, University of the Philippines-Diliman
1101 Quezon City, Philippines
jmpasia@up.edu.ph

Abstract. Multi-objective random one-bit climbers (moRBCs) are one class of stochastic local search-based algorithms that maintain a reference population of solutions to guide their search. They have been shown to perform well in solving multi-objective optimization problems. In this work, we analyze the performance of moRBCs when modified by introducing tabu moves. We also study their behavior when the selection to update the reference population and archive is replaced with a procedure that provides an alternative mechanism for preserving the diversity among the solutions. We use several MNK-landscape models as test instances and apply statistical testings to analyze the results. Our study shows that the two modifications complement each other in significantly improving moRBCs' performance especially in many-objective problems. Moreover, they can play specific roles in enhancing the convergence and spread of moRBCs.

1 Introduction

Multi-objective optimization (MO) is the process of simultaneously finding solutions to two or more, and often conflicting, objectives. It is referred to as *many*-objective optimization (MaO) if there are at least four objectives. MaO has attracted the interest of many researchers because of the poor performance of multi-objective evolutionary algorithms (MOEAs) which are known to be efficient in solving MO problems. Research studies reveal that the poor performance can be attributed to the substantially large number of solutions in the Pareto front levels when the number of objectives is high [1,2]. This makes that Pareto dominance of MOEAs coarser as they assign equal ranks to many solutions [3,4,5], thereby weakening their convergence property.

Many of the algorithms for solving MaO problems are based on modifying the MOEAs by introducing strategies such as ranking improvement [6], dimensionality reduction [7,8], objective space partitioning [9,10], and the use of preference information, scalarizing and indicator functions [11,12,13,14]. Aside from

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 182–196, 2011.

improving the conventional MOEAs, other works have focused on developing approaches that are based on the stochastic local search (SLS) paradigm. The principle of SLS is the idea behind many of the general-purpose methods like iterated local search, simulated annealing, tabu search, and variable neighborhood search [15]. In general, SLS searches for good solutions by exploiting the local knowledge provided by the definition of solution neighborhood [15]. There are several studies that demonstrate their ability to solve multi-objective problems of different domains (e.g. [16,17,18]).

Depending on the search strategy, the various SLS algorithms for multi-objective combinatorial problems can be grouped into two classes [19]. One class uses an acceptance criterion based on dominance relations and the other class is based on scalarization of the objective functions. One family of dominance-based SLS algorithms that scales up very well in MaO problems are the multi-objective random one-bit climbers (moRBCs) [20]. These algorithms follow the principle of 1+1-Evolution Strategy since they use a single parent solution to create a child. They also incorporate a reference population to sustain their search, maintain an archive that stores the efficient solutions, and limit the neighborhood structure to one-bit flip moves. They have been demonstrated to perform better than conventional multi-objective optimizers like NSGA-II [21] and SPEA2 [22] on various MNK-landscape models [4]. In fact, moRBCs (or SLS in general) are desirable many-objective optimizers especially for problem instances where the non-dominated solutions are not highly correlated in decision (genotype) space, objective (phenotype) space and between spaces since mutation operators alone have been shown to perform better than optimizers with crossover in such problem instances [4].

In this work, we propose enhancing the implementation of moRBCs by introducing a strategy that allows the search to explore for more promising solutions. We also propose improving the mechanism for updating the reference population and archive by using a sampling distribution that applies a wider dominance region. In addition, this work discusses how these methods can enhance the performance of moRBCs and demonstrates that these strategies complement each other. To evaluate the performance of moRBCs, we apply the same MNK-landscape models used in [20] as test instances.

2 Multi-objective Optimization and MNK-Landscapes

Multi-objective optimization (MO) involves simultaneously optimizing a set of two or more objective functions. Formally, it can be stated as follows:

$$\max_{\boldsymbol{x} \in \mathtt{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})) \tag{1}$$

where $\mathtt{X} \subset \mathbb{R}^N$ is the feasible space, $\boldsymbol{f} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ is a vector-valued objective function, f_i $(i = 1, 2, \ldots, M)$ denote the individual objective functions and M is the number of objectives.

In general, there are several optimal solutions, also called non-dominated solutions, to MO problems. A solution $\boldsymbol{x}$ is non-dominated if there exists no other feasible solution $\boldsymbol{y}$ such that $f_i(\boldsymbol{y}) \geq f_i(\boldsymbol{x})$, for $i = 1, 2, \ldots, M$ and $f_i(\boldsymbol{y}) > f_i(\boldsymbol{x})$ for some i. If such solution $\boldsymbol{y}$ exists, then we say that $\boldsymbol{y}$ *dominates* $\boldsymbol{x}$ ($\boldsymbol{y} \succ \boldsymbol{x}$). Other dominance relations between any two feasible solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ also exist. For example, $\boldsymbol{y}$ *weakly dominates* $\boldsymbol{x}$ ($\boldsymbol{y} \succeq \boldsymbol{x}$) if and only if $\boldsymbol{f}(\boldsymbol{y}) \geq \boldsymbol{f}(\boldsymbol{x})$ (i.e., $f_i(\boldsymbol{y}) \geq f_i(\boldsymbol{x}) \quad \forall i = 1, 2, \ldots, M$) and $\boldsymbol{y}$ ε-weakly dominates solution $\boldsymbol{x}$ if and only if $(1+\varepsilon)f(\boldsymbol{y}) \geq f(\boldsymbol{x})$. Finally, the image of the non-dominated solutions in the objective space is called the Pareto front or efficient frontier.

The MNK-landscape [4] is an extension of Kauffman's NK-landscape models of epistatic interaction [23] to multi-objective combinatorial optimization problems. Formally, the MNK-landscape is defined as a vector function mapping binary strings of length N into M real numbers $\boldsymbol{f} = (f_1, f_2, \ldots, f_M) : \mathtt{Z}^N \rightarrow \mathtt{R}^M$, where $\mathtt{Z} = \{0, 1\}$. $K = \{K_1, K_2, \ldots, K_M\}$ is a set of integers where each K_i gives the number of bits in the string that epistatically interact with each bit in the ith landscape. Each $f_i(\cdot)$ is given by the average of N functions by

$$f_i(\boldsymbol{x}) = \frac{1}{N} \sum_{j=1}^{N} f_{i,j}(x_j, z_1^{(i,j)}, z_2^{(i,j)}, \ldots, z_{K_i}^{(i,j)}) \tag{2}$$

where $f_{i,j} : \mathtt{Z}^{K_i+1} \rightarrow \mathtt{R}$ gives the fitness contribution of x_j to $f_i(\cdot)$, and $z_1^{(i,j)}, z_2^{(i,j)}, \ldots, z_{K_i}^{(i,j)}$ are the K_i bits interacting with x_j in string $\boldsymbol{x}$. Note that K defines the degree of non-linearity of the problem. Varying K_i from 0 to $N-1$ gives a family of increasing rugged multi-peaked landscapes resulting to hard-to-solve problems [4].

3 Multi-objective Random One-Bit Climbers

Multi-objective random one-bit climbers (moRBCs) begin by randomly creating a parent string $\boldsymbol{p}$ of length N and generating a random permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ of the string positions. Then, a child $\boldsymbol{c}$ is created by cloning $\boldsymbol{p}$ and flipping the bit at position π_i. Next, $\boldsymbol{c}$ is evaluated and replaces $\boldsymbol{p}$ if it satisfies the `replacement criterion`. Child creation, evaluation, and (possibly) parent replacement are repeated until all positions in π are used up. Once a parent replacement is detected, the search continues by creating a new permutation π. When no replacement is performed then a `local optimum` has been found and moRBC `restarts` the search. The algorithm stops after performing *maxeval* function evaluations.

A set of up to γ solutions not dominated by the parent and among themselves are maintained in a reference *Population* during the search process. The moRBCs perform the population `restart` by replacing the parent with one individual chosen from the *Population*. If the *Population* is empty, then the parent is replaced with a randomly generated string (this is known as *hard* `restart`). In addition, the non-dominated solutions found throughout the search are stored in an *Archive* of size σ. The procedure that updates the *Population* and the *Archive* uses diversity preserving mechanism in objective the space of NSGA-II.

Hence, the non-dominated individuals with better crowding distance are preferred when the sizes of *Population* and *Archive* exceed their limits. **Algorithm 1** provides the outline of moRBCs.

Algorithm 1. Multi-objective random one-bit climbers

```
Initialize parent $\boldsymbol{p}$ with a random string of length $N$, add $\boldsymbol{p}$ to Archive, and set
the number of evaluations counter to $t = 1$
While $(t < T)$[–generation loop–]
    Create permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ of the $N$ string positions, randomly set
    the permutation index to $i = 1$, and set local optimum to yes
    While $(i < N$ and $t < T)$ [–iteration loop–]
        - Clone the $\boldsymbol{p}$ and flip the bit at position $\pi_i$ to create child $\boldsymbol{c}$.
        - If $\boldsymbol{c}$ fulfills the replacement criterion then replace $\boldsymbol{p}$ with $\boldsymbol{c}$ and set local
          optimum to no
        - If $\boldsymbol{c}$ is not dominated by $\boldsymbol{p}$ then update Archive and Population with $\boldsymbol{c}$
        - Increment $t$ and $i$, $t = t + 1$ and $i = i + 1$
    If local optimum is yes then restart the search by replacing $\boldsymbol{p}$ with an
    individual of the Population. If Population is empty then $\boldsymbol{p}$ is initialized
    anew with a random string. Increment $t$, $t = t + 1$
Return Archive
```

In [20], different versions of moRBC were created by varying the `replacement criterion` and `restart` policy. It was revealed that moRBCs that use dominance relation as replacement criterion and apply population/hard `restart` perform very well varying the number of objectives and levels of epistatic interactions. Thus, in this work we focus on further improving this moRBC version.

4 Improvement Strategies for MoRBCs

4.1 Tabu Moves

One may observe that the moRBCs presented in **Algorithm 1** allows the search to visit quite a number of the same solutions. For example, if child $\boldsymbol{c}$ created by flipping the bit position π_i replaces the current parent and none of the succeeding children created by flipping the bits at positions π_j, $i < j \leq N$ dominates $\boldsymbol{c}$ (the new parent), then these same children will again be visited if $\boldsymbol{c}$ is determined to be a local optimum. Thus, we propose creating a new permutation π immediately after a child replaces the parent. However, it is still possible that a child will be created again in the succeeding iterations. For example, consider the series of parent solutions $\boldsymbol{c_1}(11111) \rightarrow \boldsymbol{c_2}(1111\underline{0}) \rightarrow \boldsymbol{c_3}(11\underline{0}1\underline{0})$. Then, $\boldsymbol{c_3}$ will again produce the same child (11011) created by $\boldsymbol{c_1}$. Also, $\boldsymbol{c_2}$ is a child of $\boldsymbol{c_3}$. To avoid re-visiting a solution, some of the children will not be created i.e., moves leading to these children will be considered tabu up to a certain time period (also called tabu age). Thus, a list l_c of prohibited or *tabu moves* is associated to every child $\boldsymbol{c}$ that is not dominated by the parent solution. If $i \in l_c$ then the 1-bit flip operator will not be applied to bit i.

Aside from including in the tabu list the move that lead to the child, each non-dominated child also inherits the tabu moves of its parent. For example, the tabu list of $\boldsymbol{c_3}$ does not only include "3" to avoid re-visiting $\boldsymbol{c_2}$ but also "5" to avoid cycling back to (11011). Hence, if solution $\boldsymbol{c_i}$ is obtained by flipping the j^{th} $(1 < j \leq N)$ bit of solution $\boldsymbol{c_{i-1}}$ then $l_{\boldsymbol{c_i}} = l_{\boldsymbol{c_{i-1}}} \cup \{j\}$. However, if the size of $l_{\boldsymbol{c_i}}$ exceeds the limit τ then the "oldest" move in $l_{\boldsymbol{c_{i-1}}}$ is removed. Thus, τ can be interpreted as the maximum tabu age of the moves.

4.2 Adaptive ε-Ranking

In many-objective optimization problems, the process of `restart` is called more often because the created solutions are more likely to be non-dominated by the parent solutions. Hence, the diversity of solutions preserved in the *Population* available for `restart` is a key factor to improve the performance of moRBCs. To maintain the *Population* (*Archive*) , the original moRBCs apply an updating procedure based on NSGA-II's crowding distance. In this work, we modify the updating procedure by using the adaptive ε-ranking [9].

The adaptive ε-ranking is a randomized sampling procedure that chooses, from sets of equally ranked solutions, those solutions that will be given selective advantage. By using dominance regions wider than conventional Pareto dominance, it increases selection probabilities of some of the solutions while keeping a balance effort in exploring the different regions of the objective space represented in the actual population.

The proposed sampling procedure initially forms the set $\mathcal{L}$ by copying all the extreme solutions of *Population* (*Archive*). Consequently, these extreme solutions are removed from *Population* (*Archive*). The other solutions of $\mathcal{L}$ are chosen as follows: Randomly draw a solution $\boldsymbol{y} \in Population(Archive)$ and remove all solutions in *Population* (*Archive*) that are ε-weakly dominated by $\boldsymbol{y}$ including the solution $\boldsymbol{y}$ itself. We repeat this process until *Population* (*Archive*) becomes empty. Finally, the new *Population* (*Archive*) is composed of the solutions of $\mathcal{L}$.

Similar to [9], the value of ε is adapted to ensure that the size of the updated *Population* (*Archive*) is equal to or somewhere near the value of $\gamma(\sigma)$. In contrast, we use a continuous approach in updating the value of ε instead of an incremental strategy. Thus, the value of ε in generation $t \geq 1$ is calculated using the formula

$$\varepsilon_{t+1} = \begin{cases} \varepsilon_t & \text{, if } \ |\mathcal{X}_t^b| = \sigma \\ \varepsilon_0 & \text{, if } \ |\mathcal{X}_t^b| = |\mathcal{X}_t^a| \\ \varepsilon_t \times \dfrac{|\mathcal{X}_t^b| - \sigma}{|\mathcal{X}_t^b| - |\mathcal{X}_t^a|} & \text{, otherwise} \end{cases} \tag{3}$$

where $\mathcal{X}_t^b$ and $\mathcal{X}_t^a$ are the sizes of *Population* (*Archive*) before and after applying the adaptive ε-ranking, respectively, and ε_0 is the initial value. We place a lower limit $\varepsilon_{\min}$ to the value of ε to guarantee that ε does not approach zero. If ε falls below this limit, then its value is reset to $\varepsilon_{\min}$. At this point, it is important to emphasize that the updating of the *Population* and *Archive* of the moRBCs is done only after every iteration (Lines 4-8 of **Algorithm 1**) in order to lessen the number of calls to either the crowding distance operator of NSGA-II or adaptive ε-ranking.

5 Experimental Design

5.1 Performance Metrics

Three metrics are used to assess the performance of the optimizers. First is the hypervolume $\mathcal{H}$ which measures the volume of the objective space that is weakly dominated by an approximation set or a set whose elements do not weakly dominate each other [22]. It is dependent on a reference point $\mathcal{O}$ obtained using the parameter α, i.e., $\mathcal{O} = \alpha \times (f_1^m, f_2^m, \ldots, f_M^m)$ where $f_i^m = \min\limits_{\boldsymbol{x} \in \mathcal{A}} f_i(\boldsymbol{x})$, $i = 1, 2, \ldots, M$, and $\mathcal{A}$ is the union of all approximation sets. In this work we use $\alpha = 0.99$ instead of $\alpha = 0$ as used in [20], to measure the quality of the *convergence* near the Pareto-optimal front. To calculate $\mathcal{H}$, we use the algorithm presented in [24].

The second metric is the set coverage $\mathcal{C}(X, Y)$, a binary quality indicator that gives the proportion of approximation set Y that is weakly dominated by approximation set X [25]. Thus, if A weakly dominates all members of B then $\mathcal{C}(A, B)$ is equal to 1 whereas if no member of B is weakly dominated by A, then $\mathcal{C}(A, B)$ is zero. Finally, we use the sum of maximum objective values ($\mathcal{S}_{\max}$) which measures the convergence of solutions at the extremes and around the M edges of the Pareto front [5] by providing information about their diversity or *spread*. It can be expressed as $\mathcal{S}_{\max}(X) = \sum_{i=1}^{M} \max\limits_{\boldsymbol{x} \in X} f_i(\boldsymbol{x})$.

5.2 Non-parametric Statistical Tests

In this study, we compare the performance of two or more optimizers (or configuration sets) by making statistical inferences about their underlying approximation set distributions. Thus, we perform non-parametric statistical testings called Friedman's test and Wilcoxon signed rank test on the metric values obtained by different optimizers. These tests are chosen since we wish to determine if the optimizers have identical effects without making any assumptions about the distributions of their metric values. Note that Friedman's test is applied when at least 3 optimizers are being compared while Wilcoxon signed rank test is used when only two optimizers are available. Reference [26] provides a detailed discussion about these tests.

In order for the statistical testings to give a more meaningful results, we set the initial population and random seeds the same for all optimzers. In this case, the inference relates to the ability of the different optimizers to improve the initial population [27]. We take the null hypothesis to be that the distribution of metric values are the same for all optimizers.

5.3 Test Problems and Parameter Settings

The different optimizers are evaluated using MNK-landscape models having $2 \leq M \leq 10$ objectives, $N = 100$ bits and $K = K_i \in \{0, 3, 7, 15, 25, 50\}$ $i = 1, 2, \ldots, M$ levels of epistatic interaction. There are 50 test instances for every M, K landscape model combination and each optimizer is performed once per test instance. Since the performance metrics require normalization, we use NSGA-II to generate reference solutions for the normalization process. NSGA-II uses a duplicate elimination strategy which boosts its performance especially for 2 and 3 objectives [4]. Following the parameter settings in [20], it uses 100 individuals, 2-pt crossover with 0.6 recombination rate, and $1/N$ per bit mutation rate. The values of σ and γ are fixed to 100 and *maxeval* is set to 3×10^5.

6 Results and Discussions

6.1 Effects of Tabu Ages

To analyze the effects of using tabu moves, we use different tabu ages ($\tau = 0, 2, 4, 6, 7, 10$). The zero tabu age ($\tau = 0$) means that the only modification on moRBC is the creation of the new random permutation π of the string positions every time a child replaces the parent, i.e. it does not maintain a list of tabu moves. Applying the Friedman's test, results show that there is a strong evidence that the distributions of $\mathcal{H}$ and $\mathcal{S}_{\max}$ values among the different configurations across various number of objectives and epistatic interactions are statistically different (p-values<0.05). Thus, we perform a post-hoc analysis that determines which pairs of configurations are significantly different from each other[1].

Since there are 21 pairs to analyze on each of the 54 (9×6) landscape models, we summarize the results by counting the number of landscape models where configuration X has better distribution than configuration Y at significance level of $\alpha = 0.05$. **Table 1** shows the results for both the $\mathcal{H}$ and $\mathcal{S}_{\max}$ values. It can be observed that in terms of the $\mathcal{H}$ metric, moRBCs with $2 \leq \tau \leq 10$ have better distribution than the original moRBC (Config. A) on 20 (different) landscape models on the average while A is only better in at most 4 landscape models. Overall, moRBC with $\tau = 4$ (Config. D) has the most number of landscape models (22) where it is better than A. It also has the least number of landscape models (5) where other configurations have better distributions.

In terms of $\mathcal{S}_{\max}$, one may also observe that the performance of original moRBC (Config. A) is significantly enhanced by using tabu moves especially when $6 \leq \tau \leq 10$. There is no landscape model where the distribution of $\mathcal{S}_{\max}$ values of A is better than those of Config. C,D,E,F,G ($2 \leq \tau \leq 10$), whereas there is a total of at least 44 landscapes (out of 54) where the C,D,E,F,G have better distributions than A. Overall, moRBC with $\tau = 8$ (Config. F) has the most number of landscape models (51) where it is better than A. It also has

[1] The code for this analysis is available at `http://www.r-statistics.com/2010/02/post-hoc-analysis-for-friedmans-test-r-code/`

Table 1. Number of test instances where ROW is statistically better than COL given a significance level of $\alpha = 0.05$. The last row (column) gives the sum of the column (row) entries.

Hypervolume

	A	B	C	D	E	F	G	
A	–	0	1	1	2	3	4	*11*
B	0	–	1	1	2	3	2	*9*
C	17	17	–	1	2	3	6	*46*
D	22	18	2	–	0	4	13	*59*
E	20	22	4	1	–	0	4	*51*
F	20	19	3	0	0	–	1	*43*
G	20	16	2	1	0	0	–	*39*
	99	*92*	*13*	*5*	*6*	*13*	*30*	

Sum of maximum objectives

	A	B	C	D	E	F	G	
A	–	0	0	0	0	0	0	*0*
B	0	–	0	0	0	0	0	*0*
C	44	43	–	0	0	1	6	*94*
D	48	48	5	–	0	1	9	*111*
E	50	51	7	2	–	0	2	*112*
F	51	51	14	7	1	–	1	*125*
G	50	51	13	9	6	1	–	*130*
	243	*244*	*39*	*18*	*7*	*3*	*18*	

A : original moRBC
B : moRBC ($\tau = 0$)
C : moRBC ($\tau = 2$)
D : moRBC ($\tau = 4$)
E : moRBC ($\tau = 6$)
F : moRBC ($\tau = 8$)
G : moRBC ($\tau = 10$)

the least number of landscape models (3) where other configurations have better distributions. Finally, applying a new permutation alone ($\tau = 0$) does not show significant improvement on the performance of original moRBC. This further shows that maintaining a list of prohibited moves is effective in enhancing the original moRBC.

6.2 Effects of Adaptive ε-Ranking

We refer to moRBCs with adaptive ε-ranking as moRBC_e. We set the values of ε_0 and $\varepsilon_{\min}$ to 0.05 and 0.0001, respectively. **Figure 1** plots the values of ε and the size of *Archive* before and after applying adaptive ε-ranking in one test instance where $M = 10$ and $K = 7$. Observe that the size of *Archive* fluctuates around $\sigma = 100$ thereby indicating that the adaptive selection strategy performs its function accordingly. **Table 2** summarizes the results of the Wilcoxon signed rank test on the $\mathcal{H}$ and $\mathcal{S}_{\max}$ values. Results show that the distribution of $\mathcal{H}$ values of moRBC_e are significantly better than those of original moRBC when

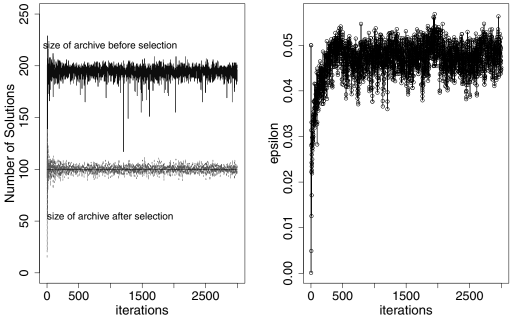


Fig. 1. Size of *Archive* before and after applying adaptive ε-ranking

Table 2. Wilcoxon *signed* rank test. The table contains the p-values with respect to alternative hypothesis that the distribution of the metric values of moRBC_e is *better* than those of original moRBC given a significance level of $\alpha = 0.05$.

Hypervolume

	$K = 0$	$K = 3$	$K = 7$	$K = 15$	$K = 35$	$K = 50$
$M = 2$	1.0000	0.9971$^{\triangleleft}$	1.0000	1.0000	1.0000	1.0000
$M = 3$	**0.0000**	**0.0002**	**0.0204**	0.9980$^{\triangleleft}$	0.9974$^{\triangleleft}$	**0.0000**
$M = 4$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$M = 5$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$M = 6$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$M = 7$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$M = 8$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$M = 9$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$M = 10$	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**

Sum of maximum objectives

	$K = 0$	$K = 3$	$K = 7$	$K = 15$	$K = 35$	$K = 50$
$M = 2$	1.0000	0.7669	1.0000	1.0000	1.0000	1.0000
$M = 3$	0.8284	1.0000$^{\triangleleft}$	0.9952$^{\triangleleft}$	0.9976$^{\triangleleft}$	0.2060	0.8158
$M = 4$	0.4511	0.9982$^{\triangleleft}$	0.9884$^{\triangleleft}$	0.9992$^{\triangleleft}$	0.9997$^{\triangleleft}$	0.1013
$M = 5$	**0.0008**	0.8304	0.9815$^{\triangleleft}$	0.9613$^{\triangleleft}$	0.9314	0.7995
$M = 6$	**0.0000**	0.1847	0.1721	0.6431	0.7220	0.0946
$M = 7$	**0.0000**	0.0823	0.1745	0.9923$^{\triangleleft}$	0.2878	0.0073
$M = 8$	**0.0000**	**0.0008**	0.1672	0.1195	0.5576	0.1004
$M = 9$	**0.0000**	**0.0000**	0.0686	**0.0387**	0.0725	**0.0053**
$M = 10$	**0.0000**	**0.0001**	**0.0000**	**0.0001**	0.2257	**0.0031**

$^{\triangleleft}$ moRBC is *better* than moRBC_e at significance level $\alpha = 0.05$

$4 \leq M \leq 10$ and for all K. When $M = 2$, the original moRBC and moRBC_e have the same distributions except when the degree of epistasis is $K = 3$. When $M = 3$, moRBC_e again has better distributions than the original moRBC but only when $K = 0, 3, 7$ and 50. These results suggest that the adaptive ε-ranking becomes more effective when applied to many-objective problems. This can be explained by the fact that there are fewer solutions in every Pareto front levels when the number of objectives is low [4], thereby limiting the potential of adaptive ε-ranking.

With regard to the $\mathcal{S}_{\max}$ metric, the adaptive ε-ranking enhances the performance of the original moRBC in limited number of landscapes. For example, it improves the original moRBC when $K = 0$ and $5 \leq M \leq 10$. Also, it generally enhances the original moRBC for landscapes where $M = 10$. However, the original moRBC has better $\mathcal{S}_{\max}$ distributions when $3 \leq M \leq 4$ and $K = 3, 7, 15$. It is also better when $M = 5$ and $K = 7, 15$, $M = 4$ and $K = 25$, and $M = 7$ and $K = 15$. In the other landscape models, no difference has been observed in their distributions.

Overall, the adaptive ε-ranking is able to improve the performance of the original moRBC. But since both the crowding distance and adaptive ε-ranking always draw the extreme solutions when sampling the population, adaptive ε-ranking has stronger positive impact on the convergence property than on the diversity or spread of the solutions of the original moRBC.

6.3 Comparison among Algorithms

We now analyze the performance of moRBCs when both tabu moves and adaptive ε-ranking are used. Since the adaptive ε-ranking greatly improves the convergence of moRBCs, we thus use $\tau = 8$ for the tabu moves because it generally emphasizes more on spread. We refer to this method as moRBC_te.

Friedman's test shows that there is statistically significant difference in the distributions of the metric values of the optimizers in all landscape models. The results of post-hoc analysis, summarized in **Table 3**, also reveal that there is no single optimizer whose distributions are better than those of the remaining optimizers. We then use a descriptive statistic to differentiate the performance quality of the optimizers. Thus, we consider optimizer OPT as the "best" optimizer in a given landscape if it fulfills the following conditions: (i) there is no optimizer whose distribution is better than OPT (ii) OPT has the best median value compared to other optimizers that satisfy condition (i). The "best" optimizer is enclosed in a box in **Table 3**.

In terms of the hypervolume, it can be observed that moRBC_e and moRBC_te are the optimizers that perform well in landscapes with $4 \leq M \leq 10$ objectives as they generate statistically better distributions. Also, moRBC_e is generally better than moRBC_te based on the two conditions discussed above. However, it is only statistically better than moRBC_te in landscapes where $M = 10$ and $K = 15, 25$. For $M = 2$, moRBC_t and moRBC_te are generally the best performing optimizers simply because adaptive ε-ranking is rarely used due to small number of solutions in the Pareto front levels. When $M = 3$, moRBC, moRBC_t and moRBC_e generally perform well.

Figure 2(a) shows the ratio $\mathcal{H}$(moRBCs)/$\mathcal{H}$(NSGAII) or the normalized hypervolume of the moRBCs varying M given $K = 7$. It can be observed that moRBC_e and moRBC_te converge better as they posted around 100% improvement (i.e. doubled) on $\mathcal{H}$ values generated by NSGA-II when $M = 4$ and 200%-300% when $7 \leq M \leq 10$. These improvements translate to better coverage of the solutions of NSGA-II by moRBCs as shown in **Fig. 3(a)**. For example, the moRBCs weakly dominates at least 40% of the solutions of NSGA-II on average while NSGA-II only covers not more than 20% of the solutions of moRBCs when $M = 3$. Also, moRBC_e and moRBC_te remarkably covers between 80% and 100% of NSGA-II when $4 \leq M \leq 5$ and around 60%-80% when $6 \leq M \leq 10$. Meanwhile, varying K and $M = 10$ yield similar results in terms of the hypervolume and coverage metrics (**Fig. 2(b)** and **3(b)**) i.e., moRBC_e and moRBC_te perform very well.

As regards the $\mathcal{S}_{\max}$ metric, **Table 3** shows that moRBC_t and moRBC_te are, in general, the optimizers that perform well in various MNK-landscapes. There is a strong evidence that the distributions of the $\mathcal{S}_{\max}$ values of at least one of them are better than those of NSGA-II, moRBC and moRBC_e for all M and $K = 7, 15, 25$ and 50. Moreover, moRBC_te is the only optimizer whose distribution of $\mathcal{S}_{\max}$ values is not outdone by the other optimizers. Applying the two conditions for determing the best optimizer in each landscape, one concludes that moRBC_te performs better than moRBC_t when $7 \leq M \leq 10$ and

Table 3. Summary of post-hoc-analysis for different performance metrics. The symbol $X \triangleright Y$ means that the distribution of optimizer X is statistically better than that of optimizer Y given a significance level of $\alpha = 0.05$. The "best" optimizers are enclosed in boxes.

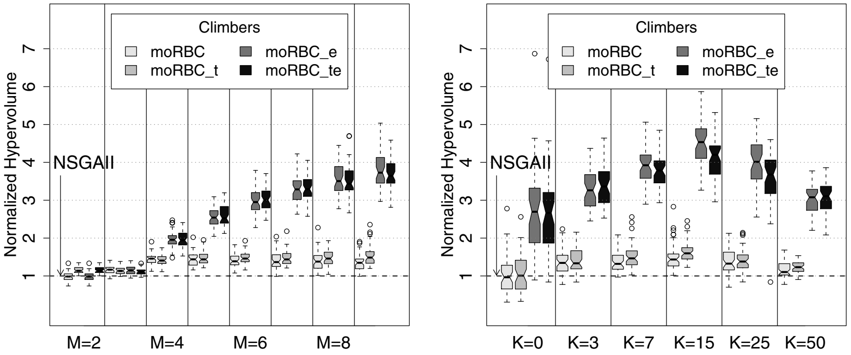
Hypervolume

	$K = 0$	$K = 3$	$K = 7$	$K = 15$	$K = 25$	$K = 50$
$M = 2$	$\boxed{\text{A}}\triangleright\text{mEN}$	$\boxed{\text{D}}\triangleright\text{AEN}$	$\boxed{\text{mD}}\triangleright\text{AEN}$	$\boxed{\text{mD}}\triangleright\text{AEN}$	$\boxed{\text{mD}}\triangleright\text{AEN}$	$\boxed{\text{mD}}\triangleright\text{AEN}$
$M = 3$	$\boxed{\text{E}}\text{m}\triangleright\text{NAD}$	$\boxed{\text{E}}\triangleright\text{mN}$	$\boxed{\text{A}}\text{DEm}\triangleright\text{N}$	$\boxed{\text{A}}\text{DE}\triangleright\text{mN}$	$\boxed{\text{D}}\text{A}\triangleright\text{EmN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$
$M = 4$	$\boxed{\text{E}}\triangleright\text{mNAD}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$
$M = 5$	$\boxed{\text{E}}\triangleright\text{mNAD}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$
$M = 6$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$
$M = 7$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$
$M = 8$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$
$M = 9$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$
$M = 10$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{m}}\text{E}\triangleright\text{ADN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$	$\boxed{\text{E}}\triangleright\text{ADmN}$	$\boxed{\text{E}}\triangleright\text{ADmN}$	$\boxed{\text{E}}\text{m}\triangleright\text{ADN}$

Sum of maximum objectives

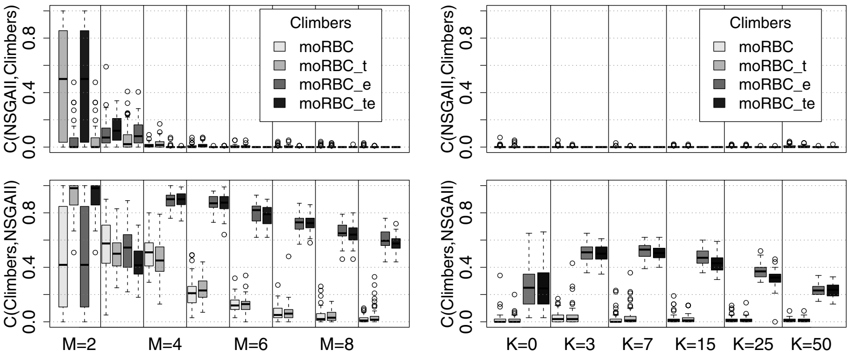
	$K = 0$	$K = 3$	$K = 7$	$K = 15$	$K = 25$	$K = 50$
$M = 2$	$\boxed{\text{ADEm}}\triangleright\text{N}$	$\boxed{\text{m}}\text{DN}\triangleright\text{AE}$	$\boxed{\text{mD}}\triangleright\text{AEN}$	$\boxed{\text{mD}}\triangleright\text{AEN}$	$\boxed{\text{Dm}}\triangleright\text{NAE}$	$\boxed{\text{Dm}}\triangleright\text{NAE}$
$M = 3$	$\boxed{\text{ADEm}}\text{N}^*$	$\boxed{\text{D}}\triangleright\text{A}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{D}}\text{m}\triangleright\text{EN}$	$\boxed{\text{D}}\text{m}\triangleright\text{NAE}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$
$M = 4$	$\boxed{\text{N}}\text{ADEm}^*$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{D}}\triangleright\text{AEN}$	$\boxed{\text{D}}\triangleright\text{AEN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$
$M = 5$	$\boxed{\text{m}}\triangleright\text{NADE}$	$\boxed{\text{m}}\text{DN}\triangleright\text{AE}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{D}}\triangleright\text{AEN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$
$M = 6$	$\boxed{\text{m}}\triangleright\text{NAD}$	$\boxed{\text{m}}\text{DN}\triangleright\text{AE}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\triangleright\text{EN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$
$M = 7$	$\boxed{\text{m}}\triangleright\text{NAD}$	$\boxed{\text{m}}\triangleright\text{ADE}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\triangleright\text{EN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$
$M = 8$	$\boxed{\text{N}}\triangleright\text{AD}$	$\boxed{\text{m}}\text{DN}\triangleright\text{AE}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\triangleright\text{EN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$
$M = 9$	$\boxed{\text{m}}\triangleright\text{NAD}$	$\boxed{\text{m}}\text{N}\triangleright\text{ADE}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\triangleright\text{EN}$	$\boxed{\text{D}}\text{m}\triangleright\text{AEN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$
$M = 10$	$\boxed{\text{m}}\triangleright\text{NADE}$	$\boxed{\text{m}}\triangleright\text{ADE}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\triangleright\text{EN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$	$\boxed{\text{m}}\text{D}\triangleright\text{AEN}$

N: NSGA-II
A : original moRBC
D : moRBC_t
E : moRBC_e
m : moRBC_te
* all optimizers are statistically the same



(a) (b)

Fig. 2. Normalized $\mathcal{H}$ (a) varying M given $K = 7$ (b) varying K given $M = 10$

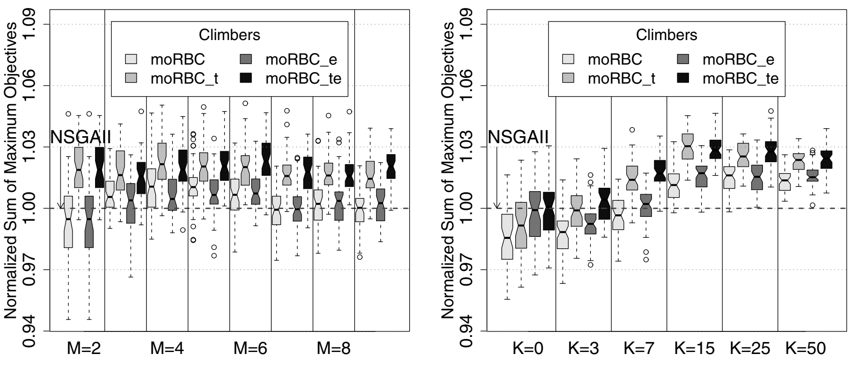


(a) (b)

Fig. 3. Coverage metric (a) varying M given $K = 7$ (b) varying K given $M = 10$

$K = 3, 7, 15, 25$ and 50 while moRBC_t excels in landscapes where $3 \leq M \leq 6$ and $K = 3, 7, 15, 25$ and 50. Both generates the same $\mathcal{S}_{\max}$ distributions when $M = 2$ and $K = 7, 15, 25$ and 50.

Figure 4(a) provides the boxplots of the normalized $\mathcal{S}_{\max}$ values varying M given $K = 7$. It is clear that moRBC_t and moRBC_te have better distributions than moRBC and moRBC_e. They also have higher $\mathcal{S}_{\max}$ values than NSGA-II by 2% on average. **Figure 4(b)** shows the normalized $\mathcal{S}_{\max}$ values varying K given $M = 10$. Again, moRBC_t and moRBC_te have better values than moRBC and moRBC_e. They also register $\mathcal{S}_{\max}$ values that are 2-3% higher than NSGA-II when $K = 7, 15, 25$ and 50.



(a) (b)

Fig. 4. Normalized $\mathcal{S}_{\max}$ (a) varying M given $K = 7$ (b) varying K given $M = 10$

In summary, our results reveal that using tabu moves significantly enhances the spread of moRBCs by generating better extreme solutions while adaptive ε-ranking greatly improves the convergence towards the Pareto-optimal front. The degree of improvement varies depending on the landscape model. Moreover, the two strategies complement each other since the same improvements in convergence and spread are realized when they are simultaneously applied.

7 Conclusions and Future Directions

In this work, we study the behavior of multi-objective random one-bit climbers (moRBCs) in MNK-landscapes when tabu moves and adaptive ε-ranking are introduced. In tabu moves, each accepted solution maintains a list of prohibited moves that may lead the search to re-visit previous solutions. Results show that using tabu moves can greatly improve the performance of moRBCs especially in terms of the quality of the spread. Also, the tabu age (τ) which determines how long the move remains tabu has different effects on the performance of using tabu moves under various landscape models.

The adaptive ε-ranking is a selection procedure that uses a wider dominance region than the conventional Pareto dominance. Our study reveals that it can also enhance the performance of moRBCs. Although in general this procedure can improve the quality of spread, it provides a more significant and remarkable improvement on the convergence behavior of the moRBCs.

Finally, our study also shows that tabu moves and adaptive ε-ranking complement each other since moRBCs that incorporate the two strategies (moRBC_te) enhance the convergence of the moRBCs with tabu moves alone (moRBC_t) and the spread of the moRBCs that only uses adaptive ε-ranking (moRBC_e). In so doing, moRBC_te maintains the quality of spread and convergence of moRBC_t and moRBC_e, respectively.

In the future, we would like to study the run-time behavior of these modified moRBCs to understand their efficiency in reaching the desired solutions. Lastly, we would like to analyze the performance of moRBCs when the one-bit neighborhood structure is extended to a more complex form.

References

1. Purshouse, R.C., Fleming, P.: Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-criterion Optimisation. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 16–30. Springer, Heidelberg (2003)
2. Khare, V., Yao, X., Deb, K.: Performance Scaling of Multi-objective Evolutionary Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 376–390. Springer, Heidelberg (2003)
3. Hughes, E.: Evolutionary many-objective optimization: Many once or one many? In: Proceedings of 2005 IEEE Congress on Evolutionary Computation, Edinburgh, pp. 222–227 (2005)
4. Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. European Journal of Operational Research 181(3), 1670–1690 (2007)
5. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: Proc. of 2008 IEEE Congress on Evolutionary Computation, Hong Kong, pp. 2424–2431 (2008)
6. Sato, H., Aguirre, H., Tanaka, K.: Controlling dominance area of solutions and its impact on the performance of mOEAs. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 5–20. Springer, Heidelberg (2007)
7. Brockhoff, D., Zitzler, E.: Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. 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. 533–542. Springer, Heidelberg (2006)
8. Saxena, D., Deb, K.: Non-linear Dimensionality Reduction Procedures for Certain Large-Dimensional Multi-objective Optimization Problems: Employing Correntropy and a Novel Maximum Variance Unfolding. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 772–787. Springer, Heidelberg (2007)
9. Aguirre, H., Tanaka, K.: Many-Objective Optimization by Space Partitioning and Adaptive ϵ-Ranking on MNK-Landscapes. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 407–422. Springer, Heidelberg (2009)
10. Jaimes, A., Aguirre, H., Tanaka, K., Coello Coello, C.: Objective Space Partitioning Using Conflict Information for Many-Objective Optimization. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 657–666. Springer, Heidelberg (2010)
11. Deb, K., Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms. In: Proc. of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 635–642. ACM, New York (2006)
12. Hughes, E.: MSOPS-II: A general purpose many-objective optimiser. In: Proc. of 2007 IEEE Congress on Evolutionary Computation, Singapore, pp. 3944–3951 (2007)

13. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)
14. Thiele, L., Miettinen, K., Korhonen, P., Molina, J.: A preference-based evolutionary algorithm for multi-objective optimization. Evolutionary Computation 17(3), 411–436 (2009)
15. Paquete, L.: Stochastic local search algorithms for multiobjective combinatorial optimization: methods and analysis. PhD thesis, FB Informatik, TU Darmstadt, Germany (2005)
16. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Transactions on Systems, Man, and Cybernetics 28(3), 392–403
17. Stützle, T.: Iterated local search for the quadratic assignment problem. European Journal of Operational Research 174, 1519–1539 (2006)
18. Pasia, J.M., Doerner, K.F., Hartl, R.F., Reimann, M.: A Population-Based Local Search for Solving a Bi-objective Vehicle Routing Problem. In: Cotta, C., van Hemert, J. (eds.) EvoCOP 2007. LNCS, vol. 4446, pp. 166–175. Springer, Heidelberg (2007)
19. Paquete, L., Stuetzle, T.: A study of stochastic local search algorithms for the biobjective qap with correlated flow matrices. European Journal of Operational Research 169(3), 943–959 (2006)
20. Aguirre, H.E., Tanaka, K.: Random bit climbers on multiobjective MNK-Landscapes: Effects of memory and population climbing. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E88-A(1), 334–345
21. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation:NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
22. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evolutionary Computation 3(4), 257–271 (1999)
23. Kauffman, S.A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, Oxford (1993)
24. Fonseca, C., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: IEEE Congress on Evolutionary Computation, Vancouver, Canada, pp. 1157–1163 (2006)
25. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
26. Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods. John Wiley & Sons, New York (1999)
27. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, Gloriastrasse 35, ETH-Zentum, 8092 Zurich, Switzerland (February 2006)

Framework for Many-Objective Test Problems with Both Simple and Complicated Pareto-Set Shapes

Dhish Kumar Saxena[1], Qingfu Zhang[2], João A. Duro[1], and Ashutosh Tiwari[1]

[1] Manufacturing Department
Cranfield University, Bedforshire, UK
[2] Department of Computing and Electronic Systems
University of Essex, Colchester, UK
{d.saxena,j.a.duro,a.tiwari}@cranfield.ac.uk
qzhang@essex.ac.uk

Abstract. Test problems have played a fundamental role in understanding the strengths and weaknesses of the existing Evolutionary Multi-objective Optimization (EMO) algorithms. A range of test problems exist which have enabled the research community to understand how the performance of EMO algorithms is affected by the geometrical shape of the *Pareto front* (PF), i.e., PF being convex, concave or mixed. However, the shapes of the *Pareto Set* (PS) of most of these test problems are rather simple (linear or quadratic), even though the real-world engineering problems are expected to have complicated PS shapes. The state-of-the-art in many-objective optimization problems (those involving four or more objectives) is rather worse. There is a dearth of test problems (even those with simple PS shapes) and the algorithms that can handle such problems. This paper proposes a framework for continuous many-objective test problems with arbitrarily prescribed PS shapes. The behavior of two popular EMO algorithms namely NSGAII and MOEA/D has also been studied for a sample of the proposed test problems. It is hoped that this paper will promote an integrated investigation of EMO algorithms for their scalability with objectives and their ability to handle complicated PS shapes with varying nature of the PF.

Keywords: Evolutionary Many-objective Optimization and Pareto-set shapes.

1 Introduction

A Multi-objective Optimization Problem (MOP) can be stated as follows:

$$\begin{aligned} \text{minimize } & F(x) = (f_1(x), \ldots, f_M(x)) \\ \text{subject to } & \qquad x \in \Omega \end{aligned} \tag{1}$$

where $\Omega = (x_1, x_2, \ldots, x_n)^T \in R^n$ is the *decision (variable) space*; R^M is the *objective space*; and $F : \Omega \rightarrow R^M$ consists of M real-valued objective functions.

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 197–211, 2011.

When Ω is a closed and connected region in R^n and all the objectives are continuous in x, the problem is referred as a *continuous MOP*. When $M \geq 4$, the problem is typically referred as a *many-objective* problem.

Let $u = (u_1, \ldots, u_M)$ and $v = (v_1, \ldots, v_M) \in R^M$ be two solutions. Assuming minimization, u is said to *dominate* v if $u_i \leq v_i$ for all $i = 1, \ldots, M$, and $u \neq v$. A point $x^\star \in \Omega$ is called *(globally) Pareto-optimal* if there is no $x \in \Omega$ such that $F(x)$ dominates $F(x^\star)$. The set of all the Pareto-optimal points is called the *Pareto set (PS)*, while the set of all the corresponding objective vectors is called the *Pareto front* [1], given by $PF = \{F(x) \in R^M | x \in PS\}$.

Evolutionary Multi-objective Optimization (EMO) algorithms aim to find a set of representative Pareto-optimal solutions in a single simulation run. Several EMO algorithms exist [2] whose behaviors and performances have been experimentally studied largely on continuous test problems [3–5] with varying PF shapes. However, the issue of how the geometrical shapes of PS affect the performance of EMO algorithms remains largely unknown owing to the fact that most of the above test problems have either linear or quadratic PS shapes. Citing two real-world instances of vehicle dynamic design [6] and power plant design [7], the authors in [8] have argued that there is no reason to believe that real-world problems would merely have linear or quadratic shapes of PS which most of the existing test problems are limited to. The issue of complicated PS has been only been rarely discussed in literature. Emphasizing the need for constructing test instances with complicated PSs, the authors in [9] have proposed a method for controlling PS shapes. However, their test instances were limited to two-objective and two-variable problems. Two relatively recent studies [10] and [11] which discuss the issue of variable linkages, indirectly touch upon the issue of complicated PSs because introducing such linkages may complicate PS shapes (but not necessarily so). The issue of variable linkages/complicated shapes has also been dealt with in [12] and [13]. In another case, the authors in [14] have examined many-objective test problems with one- and two-dimensional PS. However, the focus in these studies has again been limited to test instances with linear or quadratic shapes.

Finally, [8] provides a general class of multi-objective continuous test instances with arbitrarily prescribed PS shapes the complexity of which can be directly controlled. While the focus in [8] is limited to two and three objectives, the current paper extends the framework to four or more objectives, or in other words *many-objective* problems. The utility of this endeavor is two fold. Firstly, it holds the promise of partly remedying the dearth of test problems in the domain of evolutionary many-objective optimization. Secondly, owing to the control on the complexity of the PSs, the test instances in the proposed framework could serve as better representatives of existing/potential real-world many-objective problems.

In the remainder of this paper, first the framework for many-objective problems with arbitrarily prescribed PS shapes is proposed. This is followed by the study of the performance of two well known EMO algorithms, namely NSGAII [15] and MOEA/D [16], on a sample of the proposed test problems.

2 Brief Background for Evolutionary Many-Objective Optimization

Many-objective optimization problems are those which involve four or more objectives. Research in evolutionary many-objective optimization has revealed that the existing EMO algorithms scale poorly with the number of objectives [17,18]. The poor scalability of the existing EMO algorithms has been found to relate to *dominance resistance* [19]; *active diversity maintenance* [20] and unknown effect of *recombination parameters* [19,20]. The high computational cost and difficulty in visualization of many-objective space add to the challenge. While one set of approach aims to counter the above difficulties by use of preference-ordering information [21], another set of approach explores the possibility of problem simplification. The latter, also referred as the *objective reduction approach* attempts to eliminate objectives that are not essential to describe the PF. In other words, given an M-objective problem, the objective reduction approach aims to identify those m ($m < M$) objectives which describe the complete PF. Such m objectives are referred as the *essential* or *critical* objectives while those which could be discarded are referred as *redundant* objectives.

The only scalable test problems belong to the DTLZ [5] or the WFG [10] test suite. However, in both these test suites, the complexity of the PS can not be controlled. It is fair to believe that testing of existing EMO algorithms on problems with varying complexity of the PS shapes will be instrumental in better understanding of the causality of their failure. Only then could the remedial approaches be proposed in future. This paper aims to propose a framework for many-objective problems with arbitrarily prescribed PS shapes.

3 Framework for Many-Objective Problems with Arbitrarily Prescribed PS Shapes

The framework being proposed here for many-objective problems is an extension of that introduced in [8] for two- and three-objective problems.

In the proposed framework, M denotes the number of objectives; m denotes the dimension of the PF ($m \leq M$); and n denotes the number of decision variables. The decision space be given by:

$$\Omega = \prod_{i=1}^{n} [a_i, b_i] \subset \mathbb{R}^n, \tag{2}$$

where $-\infty < a_i < b_i < +\infty$ for all $i = 1, \ldots, n$.

Furthermore, the M objectives (to be minimized) take the following form:

$$f_i(x) = \alpha_i(x_I) + \beta_i(x_{II} - g(x_I)) \ \forall i = 1, \ldots, M \tag{3}$$

where

- $x = (x_1, \ldots, x_n) \in \Omega$; and, x_I and x_{II} are two subvectors of x, such that $x_I = (x_1, \ldots, x_m)$ and $x_{II} = (x_{m+1}, \ldots, x_n)$.
- α_i $(i = 1, \ldots, M)$: functions from $\prod_{i=1}^{m}[a_i, b_i]$ to $\mathbb{R}$.
- β_i $(i = 1, \ldots, M)$: functions from $\mathbb{R}^{n-m}$ to $\mathbb{R}^+$.
- g : a function from $\prod_{i=1}^{m}[a_i, b_i]$ to $\prod_{i=m+1}^{n}[a_i, b_i]$.

The PF and PS of the above framework is governed by the following theorem[1]:

Theorem 1. Suppose that

[**i**] $\beta_i(z) = 0$ for all $i = 1, \ldots, m$ iff $z = 0$;
[**ii**] The PS of the following problem:

$$\begin{array}{l} \text{minimize } (\alpha_1(x_I), \ldots, \alpha_M(x_I)) \\ \text{subject to } x_I \in \prod_{i=1}^{m}[a_i, b_i] \end{array} \tag{4}$$

is $E \subset \prod_{i=1}^{m}[a_i, b_i]$.

Then the PS of the framework defined by (2) and (3) is

$$x_{II} = g(x_I), \quad x_I \in E$$

and its PF is the same as that of (4), i.e.

$$\{(\alpha_1(x_I), \ldots, \alpha_M(x_I)) | x_I \in E\}.$$

It may be noted that:

- If the dimensionality of E is m, then the PS and PF of the test problem will also be m-D.
- While the α_is determine the PF of the test problem, the PS is determined by g functions.
- The difficulty of convergence is controlled by β_i functions. If $\sum_{i=1}^{m} \beta_i$ has many local minima, then the test problem may have many local Pareto optimal solutions.
- In comparison to DTLZ [5] or the WFG [10], this framework is similar to the extent that it uses component functions for defining the PF and introducing multimodality. However, the framework holds the advantage in terms of its controllability of the complexity of PS.
- For an M-objective problem with an m-dimensional PF, β_is are functions from $\mathbb{R}^{n-m}$ to $\mathbb{R}^+$. Hence, let $x_{II} - g(x_I)$ be represented by $y_{m+1:n}$, following which (3) could be compacted to the following form:

$$f_i(x) = \alpha_i(x_I) + \beta_i(y_{m+1:n}), \; \forall i = 1, \ldots, M \tag{5}$$

[1] The proof of this theorem can be found in [8].

3.1 Modular Approach to Test Instances

This section proposes a modular approach of composing test instances with varying complexity of the PF and PS. It involves the following steps:

1. Decide for the number of objectives (M), the desired dimension of the PF (m) and the number of decision variables (n).
2. Define the β function that controls the degree of convergence. In this paper, the generalized form of the β function used in [8] has been proposed. Reference to Equations 3 and 5 reveals that $X_{II} - g(x_I) = (x_{m+1} - g_{m+1}, \ldots, x_n - g_n)$ is represented by $y_{m+1:n}$, i.e., $y \in \mathbb{R}^{n-m}$. Given this, the β functions are defined as follows:

$$\begin{aligned}
\beta_1(y_{m+1:n}) &= \tfrac{2}{|J_1|} \textstyle\sum_{j \in J_1} y_j^2, \\
&\quad \text{where } J_1 = \{j | m+1 \leq j \leq n, \text{ and } j-1 \text{ is a multiple of m+1}\} \\
\beta_2(y_{m+1:n}) &= \tfrac{2}{|J_2|} \textstyle\sum_{j \in J_2} y_j^2, \\
&\quad \text{where } J_2 = \{j | m+1 \leq j \leq n, \text{ and } j-2 \text{ is a multiple of m+1}\} \\
\vdots \qquad & \quad \vdots \\
\beta_m(y_{m+1:n}) &= \tfrac{2}{|J_m|} \textstyle\sum_{j \in J_m} y_j^2, \\
&\quad \text{where } J_m = \{j | m+1 \leq j \leq n, \text{ and } j-m \text{ is a multiple of m+1}\} \\
\beta_{m+1}(y_{m+1:n}) &= \tfrac{2}{|J_{m+1}|} \textstyle\sum_{j \in J_{m+1}} y_j^2, \\
&\quad \text{where } J_{m+1} = \{j | m+1 \leq j \leq n, \text{ and } j \text{ is a multiple of m+1}\} \\
\beta_k(y_{m+1:n}) &= A_k \textstyle\sum_{j=m+2}^{n} a_{kj} y_j^2, \forall k = m+2, \ldots, M, \\
&\quad \text{where } A_k, a_{ij} > 0 \text{ are randomly generated constants.}
\end{aligned} \tag{6}$$

3. Corresponding to a defined β function (as above), create different test instances based on Equation 5 by picking different combinations of S_α and S_y, a few of which are presented in Tables 1 and 2, respectively.

4 EMO Algorithms Used and the Parameter Settings

The EMO algorithms considered are namely NSGAII [15] and MOEA/D [16]. MOEA/D is a decomposition based algorithm that is able to use different approaches for converting the problem of approximation of the PF into a number of scalar optimization problems. The approaches considered in the simulations are (i) weighted sum [22], (ii) Tchebycheff [22], and (iii) Normal Boundary Intersection (NBI) [23]. The probability of crossover for NSGAII is 0.9 while probability of mutation is 0.1. For MOEA/D the probability of selecting mating parents from neighboring subproblems is 0.9 and the number of nearest neighboring subproblems "*niche*" is 20. The MOEA/D weight vectors, $\lambda^1, \ldots, \lambda^N$, are controlled by a parameter H, selected as 6 for all test instances. Since each individual weight vector takes the form $\{0/H, 1/H, \ldots, H/H\}$, the number of weight vectors is determined by

$$N = C_{M-1}^{H+M-1}.$$

Table 1. The proposed sets of $\alpha(x_I)$ (Equation 5)

S_α	Set Composition
For $m = 1$ implying a one-dimensional PF: Variable bounds are $[0, 1]$	
S_{α_1}	$\alpha_1(x_I) = x_1$ $\alpha_2(x_I) = 1 - \sqrt{x_1}$ $\alpha_k(x_I) = (\frac{k-2}{M} - \sqrt{x_1})^2 \ \ \forall k = 3, 4, \dots, M.$
S_{α_2}	$\alpha_1(x_I) = x_1$ $\alpha_2(x_I) = 1 - sin(2\pi x_1)$ $\alpha_k(x_1) = [x_1 - 0.25 \times \frac{(k-1)}{M}]^2 \ \ \forall k = 3, 4, \dots, M.$
S_{α_3}	$\alpha_1(x_I) = x_1$ $\alpha_2(x_I) = 1 - sin(2\pi x_1)$ $\alpha_k(x_1) = \begin{cases} \psi_1(x_1) = [x_1 - 0.25 \times \frac{(k-1)}{M}]^2, \text{ if } 0 \leq x_1 \leq 0.75 \\ \psi_2(x_1) = A + (x_1 - 0.75 - 0.25 \times \frac{(k-1)}{M})^2, \text{ otherwise} \\ \text{where } A : \psi_1(0.75) = \psi_2(0.75) \ \ \forall k = 3, 4, \dots, M. \end{cases}$
For $m = 2$ implying a two-dimensional PF: Variable bounds are $[0, 1]^2$	
S_{α_4}	$\alpha_1(x_I) = \sin(0.5x_1\pi)$ $\alpha_2(x_I) = \cos(0.5x_1\pi)\sin(0.5x_2\pi)$ $\alpha_3(x_I) = \cos(0.5x_1\pi)\cos(0.5x_2\pi)$ $\alpha_k(x_I) = c_1 \times \{\alpha_1(x_I)\}^{p_1} + c_2 \times \{\alpha_2(x_I)\}^{p_2} + c_3 \times \{\alpha_3(x_I)\}^{p_3} \ \forall k = 4, 5, \dots, M$ where $c_i, p_i > 0$ are arbitrary constants for $i = 1, 2, 3$

Table 2. The proposed sets of y_j (Equation 5)

S_y	Set Composition	Variable bounds for x_j
For $m = 1$ implying a one-dimensional PS		
S_{y_1}	$x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}$	$[0, 1]^{n-1}$
S_{y_2}	$x_j - \sin(6\pi x_1 + \frac{j\pi}{n})$	$[-1, 1]^{n-1}$
S_{y_3}	$x_j - \begin{cases} 0.8x_1\cos(6\pi x_1 + \frac{j\pi}{n}) \ j \in J_1 \\ 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n}) \ j \in J_2 \end{cases}$	$[-1, 1]^{n-1}$
S_{y_4}	$x_j - \begin{cases} 0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}) \quad j \in J_1 \\ 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n}) \ j \in J_2 \end{cases}$	$[-1, 1]^{n-1}$
S_{y_5}	$x_j - \begin{cases} 0.3x_1(x_1\cos(4\theta) + 2)\cos(\theta) \ j \in J_1 \\ 0.3x_1(x_1\cos(4\theta) + 2)\sin(\theta) \ j \in J_2 \\ \text{where } \theta = 6\pi x_1 + \frac{j\pi}{n} \end{cases}$	$[-1, 1]^{n-1}$
For $m = 2$ implying a two-dimensional PS		
S_{y_6}	$x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})$	$[-2, 2]^{n-2}$

The number of objectives (M) selected for all test problems is 5, therefore, the MOEA/D population size is 210[2] and for NSGAII is 212[3]. For statistical purposes, all the simulations performed have been reproduced using 20 evenly spaced seeds between 0 and 1.

Table 3. Construction of sample test problems using the proposed framework. New problems can be created using different combinations of S_α and S_y.

	S_α					
	One dimensional PF (1D-PF)			Two dimensional PF (2D-PF)		
S_y	S_{α_1}	S_{α_2}	S_{α_3}	S_{α_4}[a]	S_{α_4}[b]	PS
S_{y_1}	Problem-1					1-D
S_{y_2}		Problem-2				1-D
S_{y_3}			Problem-3			1-D
S_{y_4}			Problem-4			1-D
S_{y_5}	Problem-5					1-D
S_{y_6}				Problem-6	Problem-7	2-D

[a] Linear: $p_1 = p_2 = p_3 = 1.0$.
[b] Non-linear: $c_1 = c_2 = c_3 = 1.0$.

5 Experimental Results

This section presents the results corresponding to the 5-objective instance of the test problems created by using the proposed framework. The construction of these test problems is presented in Table 3. For these problems, the performance of two EMO algorithms, namely NSGAII [15] and MOEA/D [16] has been studied. The results are reported in Table 4 where the *inverted generational distance* (IGD) [24] is used as the quality indicator. To observe the effect of variable scaling, the 1D-PF test problems (P1 to P5) have been solved with 4 and 10 variables respectively, while the 2D-PF test problems (P6 and P7) are solved with 5 and 10 variables respectively. Furthermore, to observe the effect that the number of generations of the EMO algorithms' run have on their performance, each test instances has been solved for 200 and 1000 generations respectively. From Table 4, it can be seen that:

1. On an average, MOEA/D with Tchebycheff approach is the best performer for the 1D-PF test problems, while the performance of MOEA/D with NBI approach is better than other methods for the 2D-PF test instances. NSGAII is slightly better than the MOEA/D counterparts for 4-variable instances of a few problems (such as P3 and P4).

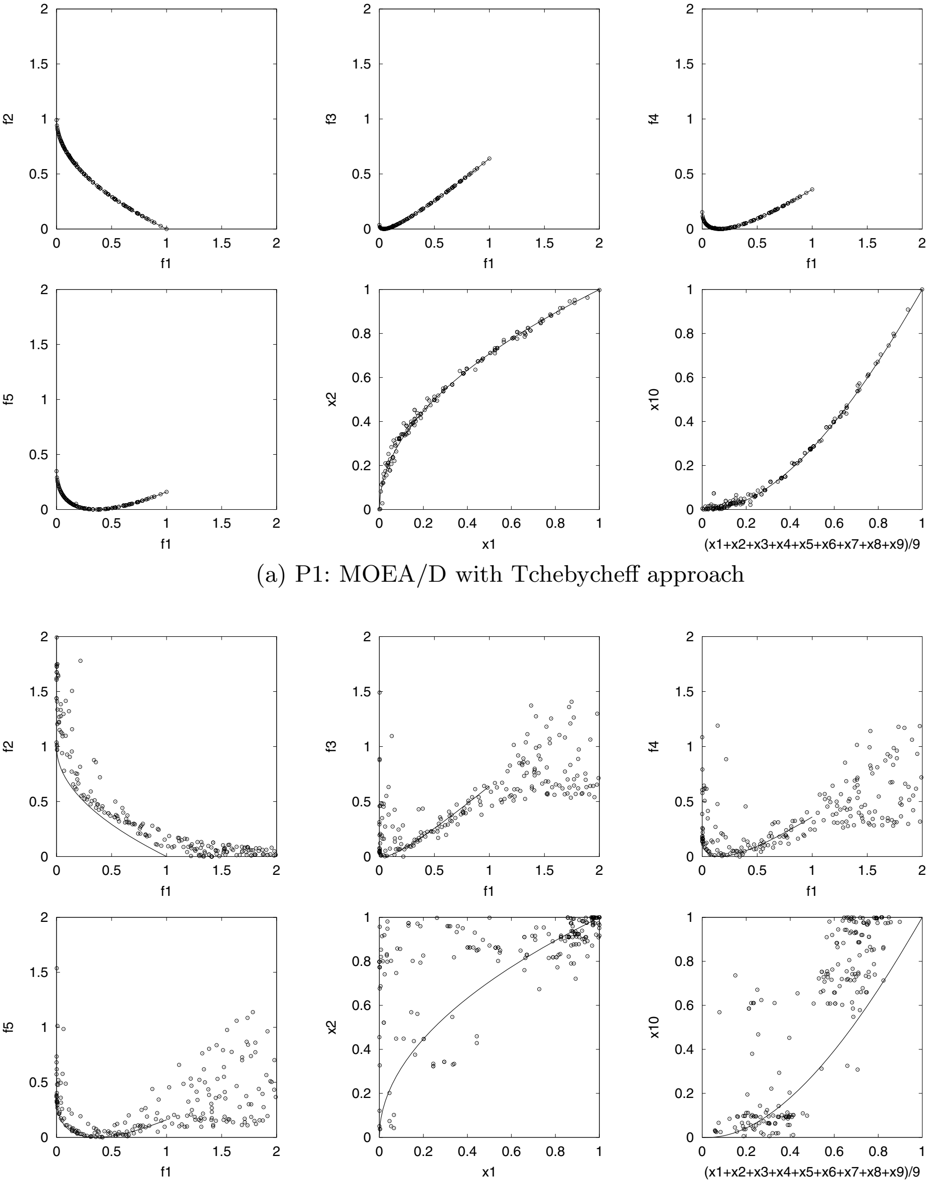
[2] Equals to the number of uniform weight vectors generated. $N = C_{5-1}^{6+5-1} = 210$.
[3] Governed by multiplicity by 4 and closeness to MOEA/D population size.

2. The performance of all the methods deteriorates with an increase in the number of variables, while it improves with an increase in the number of generations. Notably,the fall in the performance of NSGAII is more significant than other methods, as the number of variables increase.

Table 4. IGD results (averaged for 20 runs) for the 5-objective test problems constructed using the proposed framework. The best results are underlined. MOEA/D approaches include: (a) weighted sum; (b) Tchebycheff and (c) NBI.

Problem	n	Gen	MOEA/D(a)	MOEA/D(b)	MOEA/D(c)	NSGAII
P1	4	200	0.0174±0.0008	0.0099±0.0003	0.0613±0.0037	0.0387±0.0102
		1000	0.0192±0.0005	0.0098±0.0001	0.0610±0.0022	0.0328±0.0063
	10	200	0.0167±0.0011	0.0102±0.0002	0.0851±0.0095	0.0783±0.0167
		1000	0.0189±0.0007	0.0099±0.0001	0.0784±0.0074	0.0860±0.0251
P2	4	200	0.0192±0.0070	0.0068±0.0004	0.1165±0.0064	0.0064±0.0012
		1000	0.0093±0.0021	0.0064±0.0002	0.1160±0.0052	0.0102±0.0033
	10	200	0.0788±0.0261	0.0201±0.0090	0.1573±0.0326	0.1193±0.0321
		1000	0.0461±0.0207	0.0088±0.0009	0.1539±0.0286	0.1588±0.0862
P3	4	200	0.0071±0.0003	0.0065±0.0002	0.1149±0.0050	0.0046±0.0009
		1000	0.0078±0.0002	0.0062±0.0002	0.1142±0.0027	0.0045±0.0016
	10	200	0.0103±0.0017	0.0085±0.0009	0.1322±0.0071	0.0384±0.0066
		1000	0.0072±0.0005	0.0071±0.0004	0.1239±0.0029	0.0387±0.0089
P4	4	200	0.0069±0.0004	0.0064±0.0002	0.1142±0.0038	0.0042±0.0005
		1000	0.0078±0.0002	0.0062±0.0002	0.1131±0.0022	0.0039±0.0007
	10	200	0.0092±0.0012	0.0078±0.0006	0.1228±0.0057	0.0278±0.0045
		1000	0.0069±0.0004	0.0069±0.0003	0.1197±0.0026	0.0274±0.0087
P5	4	200	0.0450±0.0109	0.0182±0.0048	0.1094±0.0331	0.0196±0.0034
		1000	0.0384±0.0104	0.0140±0.0026	0.0863±0.0191	0.0195±0.0020
	10	200	0.0666±0.0060	0.0441±0.0110	0.1945±0.0282	0.0610±0.0077
		1000	0.0623±0.0063	0.0275±0.0043	0.1771±0.0284	0.0509±0.0078
P6	5	200	0.5923±0.0321	0.3372±0.0164	0.3280±0.0085	0.3403±0.0399
		1000	0.6111±0.0454	0.3315±0.0193	0.3356±0.0059	0.2807±0.0163
	10	200	0.6200±0.0685	0.3580±0.0226	0.3595±0.0457	0.7581±0.2329
		1000	0.6244±0.0740	0.3514±0.0139	0.3461±0.0043	0.4735±0.1374
P7	5	200	0.7554±0.0332	0.7395±0.0088	0.6117±0.0046	0.7537±0.0340
		1000	0.7571±0.0330	0.7400±0.0043	0.5976±0.0020	0.7321±0.0207
	10	200	0.8088±0.0672	0.7558±0.0470	0.7072±0.0818	1.2089±0.2628
		1000	0.7874±0.0650	0.7475±0.0172	0.6651±0.0870	0.8411±0.0832

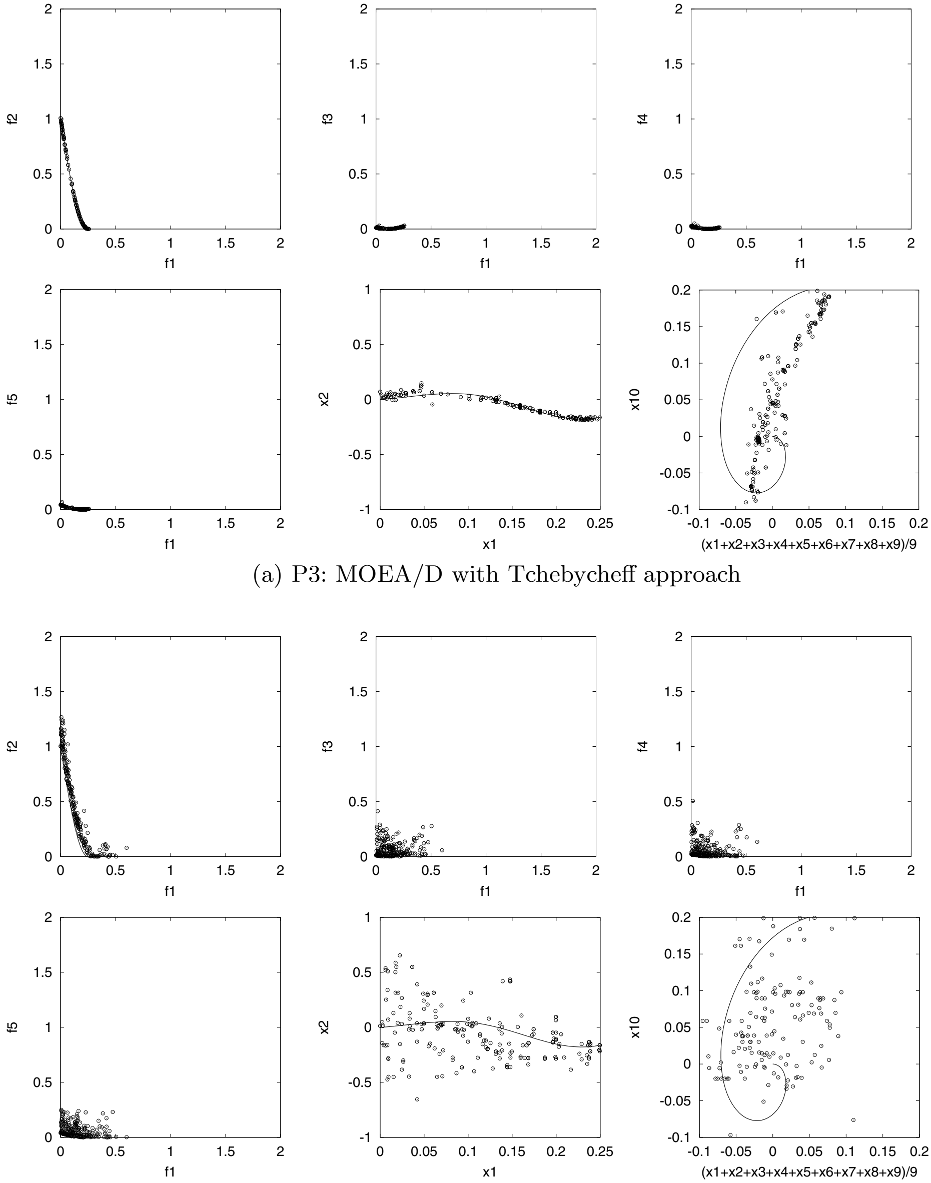
To facilitate a better understanding, the results presented in Table 4 are also graphically presented below for a few problems. The latter correspond to (i) 10-variable instances (more difficult than 4-variable instances) for 1D-PF problems, and 5-variable instance for a 2D-PF problem, (ii) 1000 generations (the results being better than those with 200 generations), and (iii) NSGAII along with the MOEA/D corresponding to the approach that offers the best results (implying MOEA/D with Tchebycheff approach for 1D-PF problems and MOEA/D with NBI approach for 2D-PF problems).



(a) P1: MOEA/D with Tchebycheff approach

(b) P1: NSGAII

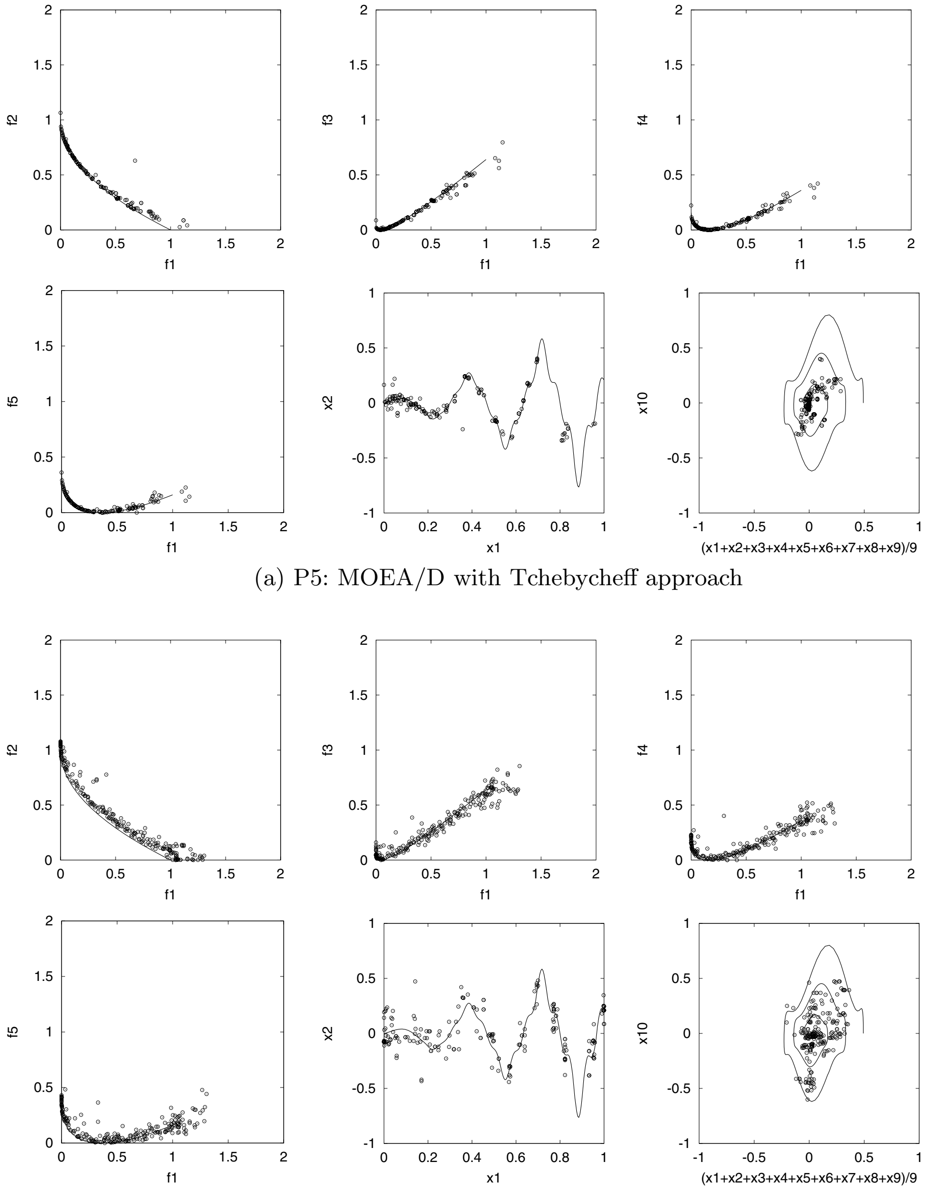
Fig. 1. Problem P1: Best approximations obtained after 1000 generations for 5-objective and 10-variable version. P1 is composed of β (Equation 6); α_is as in S_{α_1} (Table 1) and y_j as in S_{y_1} (Table 2).



(a) P3: MOEA/D with Tchebycheff approach

(b) P3: NSGAII

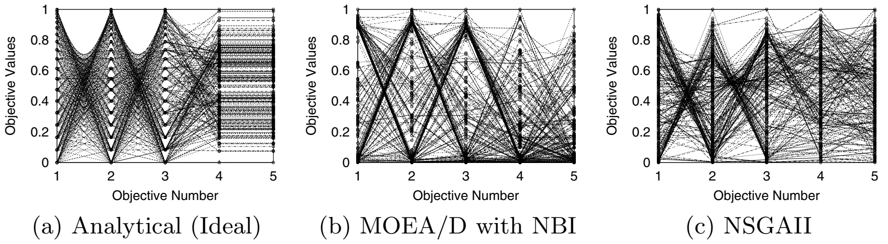
Fig. 2. Problem P3: Best approximations obtained after 1000 generations for 5-objective and 10-variable version. P3 is composed of β (Equation 6); α_is as in S_{α_3} (Table 1) and y_j as in S_{y_3} (Table 2).



(a) P5: MOEA/D with Tchebycheff approach

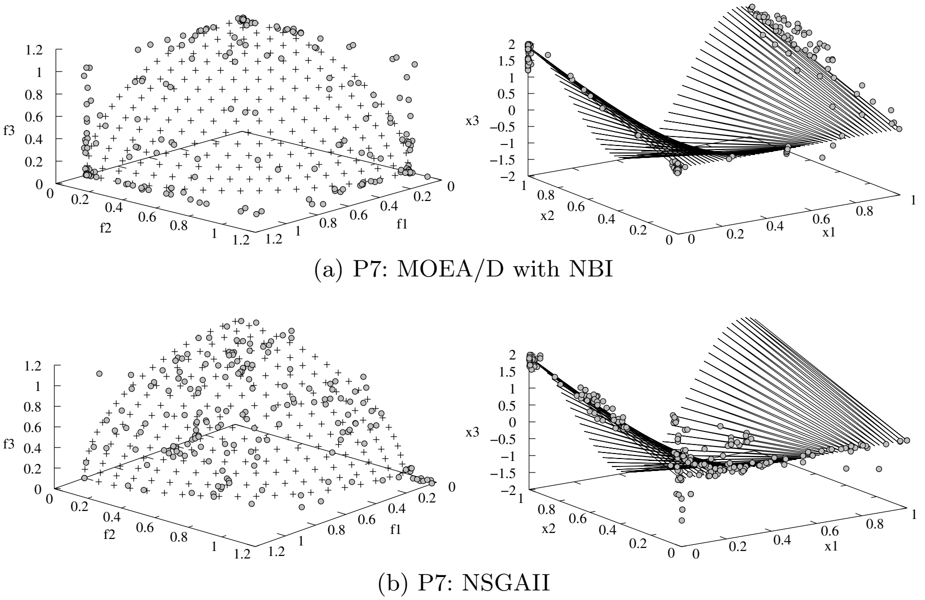
(b) P5: NSGAII

Fig. 3. Problem P5: Best approximations obtained after 1000 generations for 5-objective and 10-variable version. P5 is composed of β (Equation 6); α_is as in S_{α_1} (Table 1) and y_j as in S_{y_5} (Table 2).



(a) Analytical (Ideal) (b) MOEA/D with NBI (c) NSGAII

Fig. 4. Problem P7: Parallel coordinate plots after 1000 generations for 5-objective and 5-variable version from a single run



(a) P7: MOEA/D with NBI

(b) P7: NSGAII

Fig. 5. Problem P7: Best approximations obtained after 1000 generations for 5-objective and 5-variable version. P7 is composed of β (Equation 6); α_is as in S_{α_4} (Table 1) and y_j as in S_{y_6} (Table 2).

Based on the graphical results presented in Figures 1,2,3, 4 and 5, the following observations can be made:

1. For P1 and P3, the Table 4 suggests significant superiority of MOEA/D (with Tchebycheff approach) over NSGAII. This is validated by the Figures 1 and 2, where both the PF and PS can be seen to be better approximated by MOEA/D than NSGAII.
2. For P5, it can be noted that the superioirty of MOEA/D (with Tchebycheff approach) over NSGAII is only marginal. This is again validated by Figure 3, where the NSGAII approximations seem to be closer to those of MOEA/D.

3. The PS shape complexity interferes with the performance of the algorithms. For instance, MOEA/D could better approximate the quadratic PS as in P1 (Figure 1a) as compared to a PS with non-linear shape, as in P5 (Figure 3a).
4. For a 2D-PF problem such as P7, it can be seen that the approximation of both the PF and the PS is poor for both MOEA/D and NSGAII. It can be seen in Figure 4 that both these methods failed to capture the non-conflicting relationship between the last two objectives.
5. It may be noted that the test problems constructed in this paper are such that for an M objective problem with m-dimensional front, only the first $m+1$ objectives are in total conflict with each other while the rest are either partly conflicting or non-conflicting with others. This feature is evident in Figures 1,2 and 3 corresponding to 1D-PF problems where only the first two objectives are in total conflcit. For a 2D-PF problem like P7, it implies that the first three objectives are in total conflict and these are plotted in Figure 5 to further help compare the MOEA/D and NSGAII approximations.

6 Conclusions

This paper presented a framework for constructing continuous many-objective test problems with arbitrarily prescribed PS shapes. Hopefully, this will facilitate the performance analysis of EMO algorithms in dealing with both complicated PS shapes and scaled number of objectives, simultaneously. Sample test problems constructed using the proposed framework have been used to study the performance of NSGAII and MOEA/D. These two algorithms were chosen as they are based on different techniques, in that while the former is a Pareto-ranking based method, the latter is a decomposition-based method. As a first step, the analysis has been focussed on many-objective problems with lower-dimensional Pareto front. In future work, the authors will endeavour to extend this study for problems with high-dimensional Pareto front, including real-world problems with such characteristics. The authors also intend to analyse the performance of EMO algorithms with different underlying techniques, such as in AMOSA [25]. Lastly, it may be noted that the problems proposed in this paper are such that the dimension of the Pareto front coincides with the dimension of the Pareto set. In future, the authors will also endeavour to extend the proposed framework to accomodate problems in which the dimension of the Pareto set is higher than the dimension of the Pareto front.

References

1. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
2. Kalyanmoy, D.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., New York (2001)
3. Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation 7, 205–230 (1999)

4. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)
5. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. In: Abraham, A., Jain, R., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, pp. 105–145. Springer, Heidelberg (2005)
6. Kasprzak, E., Lewis, K.: An approach to facilitate decision trade-offs in pareto solution sets. Journal of Engineering Valuation and Cost Analysis 3, 173–187 (2000)
7. Hillermeier, C.: Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach. Birkhäuser-Verlag, Basel (2000) ISBN 978-3764364984
8. Li, H., Zhang, Q.: Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13, 284–302 (2009)
9. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 792–802. Springer, Heidelberg (2004)
10. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10, 477–506 (2006)
11. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 1141–1148 (2006)
12. Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. 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. 583–592. Springer, Heidelberg (2006)
13. Zhang, Q., Zhou, A., Jin, Y.: Rm-meda: A regularity model-based multiobjective estimation of distribution algorithms. IEEE Transactions on Evolutionary Computation 21, 41–63 (2008)
14. Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., Nojima, Y.: Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 91–100. Springer, Heidelberg (2010)
15. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
16. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11, 712–731 (2007)
17. Hughes, E.: Evolutionary many-objective optimisation: many once or one many? In: IEEE Congress on Evolutionary Computation, vol. 1, pp. 222–227. IEEE, Los Alamitos (2005)
18. Knowles, J., Corne, D.W.: Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 757–771. Springer, Heidelberg (2007)
19. Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. IEEE Transactions on Evolutionary Computation 11, 770–784 (2007)

20. Purshouse, R.C., Fleming, P.J.: Evolutionary many-objective optimization: An exploratory analysis. In: IEEE Congress on Evolutionary Computation, pp. 2066–2073 (2003)
21. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: IEEE Congress on Evolutionary Computation, pp. 2424–2431 (2008)
22. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Dordrecht (1999) ISBN 0-7923-8278-1
23. Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J. on Optimization 8, 631–657 (1998)
24. Veldhuizen, D.A.V., Lamont, G.B.: Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, 1998, Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., Wright-Patterson, AFB, OH (1998)
25. Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K.: A simulated annealing-based multiobjective optimization algorithm: Amosa. IEEE Transactions on Evolutionary Computation 12, 269–283 (2008)

A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE

Karthik Sindhya[1], Ana Belen Ruiz[2], and Kaisa Miettinen[1]

[1] Department of Mathematical Information Technology, P.O. Box 35 (Agora),
FI-40014 University of Jyväskylä, Finland
karthik.sindhya@jyu.fi
kaisa.miettinen@jyu.fi

[2] Department of Applied Economics (Mathematics)
University of Malaga
C/ Ejido 6, 29071, Malaga, Spain
abruiz@uma.es

Abstract. This paper describes a new Preference-based Interactive Evolutionary (PIE) algorithm for multi-objective optimization which exploits the advantages of both evolutionary algorithms and multiple criteria decision making approaches. Our algorithm uses achievement scalarizing functions and the potential of population based evolutionary algorithms to help the decision maker to direct the search towards the desired Pareto optimal solution. Starting from an approximated nadir point, the PIE algorithm improves progressively the objective function values of a solution by finding a better solution at each iteration that improves the previous one. The decision maker decides from which solution, in which direction, and at what distance from the Pareto front to find the next solution. Thus, the PIE algorithm is guided interactively by the decision maker. A flexible approach is obtained with the use of archive sets to store all the solutions generated during an evolutionary algorithm's run, as it allows the decision maker to freely navigate and inspect previous solutions if needed. The PIE algorithm is demonstrated using a pollution monitoring station problem and shown to be effective in helping the decision maker to find a solution that satisfies her/his preferences.

1 Introduction

Optimization problems arising in all areas of science, engineering, and economics often involve multiple conflicting objectives. Such problems are multi-objective optimization problems. A solution to a multi-objective optimization problem desirably belongs to a set of mathematically equally good compromise solutions with different trade-offs, called Pareto optimal solutions. In addition, it should satisfy the preferences of a decision maker (DM). The set of Pareto optimal solutions in the objective space is called Pareto front. Different solution approaches to solve multi-objective optimization problems exist in the literature, among

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 212–225, 2011.

them approaches of Evolutionary Multi-objective Optimization (EMO) [3,4] and Multiple Criteria Decision Making (MCDM) [2,16] are commonly used.

Evolutionary multi-objective optimization algorithms deal with a population of solutions and try to find a set of well-distributed non-dominated solutions that approximate the entire Pareto front. For the past two decades, several EMO algorithms have been proposed [3,4] and they have been shown to be very versatile in handling different types of variables and objectives. Compared to EMO, the main goal in MCDM is to aid the DM in finding a satisfying solution according to her/his preference information. Commonly, a multi-objective problem is scalarized into a single-objective problem taking into account the DM's preference information, which can be expressed e.g. with trade-offs, pairwise comparisons, aspiration levels, reference points and/or classification of the objective functions. The MCDM approaches can be broadly classified in to *a priori*, *a posteriori* and *interactive* methods, depending on when the preference information is asked from the DM [15]. For a detailed description of different MCDM approaches see [16]. Although only the most interesting Pareto optimal solutions are found by MCDM approaches, which saves computational time, their success in solving real world problems depends on the method used to solve scalarized single objective optimization problems (e.g. whether the solution is globally or locally optimal).

Advantages of both EMO and MCDM approaches can be utilized by hybridizing them. At least two possible ways can be identified: "evolutionary algorithm in MCDM" and "MCDM in EMO" approaches. Utilizing (single objective) population based approaches such as evolutionary algorithms to solve the scalarized functions formulated in MCDM approaches (we call this approach as "evolutionary algorithms in MCDM") has received some attention (see e.g. [12,17]). In this way, nonconvexity or discontinuity in the problem or possible binary or integer variables can be handled in a straightforward way. On the other hand, in "MCDM in EMO" approach based algorithms (see e.g. [1,8,9,21]), instead of approximating the entire Pareto front, the main goal is to find a solution or a set of solutions which approximates a region of the Pareto front that is interesting to the DM. Thus the DM gets a preferred set of solutions and (s)he can choose the most satisfactory solution among them.

On one hand, showing solutions only in the preferred region of the Pareto front allows the DM to inspect only a handful of solutions and can gain insights about the trade-offs between the objectives in the desired region and the final solution can be selected from a knowledgeable position. On the other hand, such an approach limits the interactions of the DM with the algorithm. In such methods, the effort is concentrated in looking just for the solutions in the preferred region(s) of the Pareto front, so a faster algorithm is obtained as irrelevant solutions are discarded, but the full potential of a population based approach to help the DM to freely navigate, inspect, and restate preferences during the solution process is yet to be fully explored. For us, any approach that adapts to the DM's desires as much as possible and that allows her/him to modify the preferences during an algorithm's run time is the best choice enabling learning of the problem and one's preferences. This is the basic idea of the Preference based Interactive Evolutionary algorithm for multi-objective optimization (PIE) that we propose in this paper.

In the PIE algorithm, the search for the most preferred solution satisfying the preference information of the DM follows nearly the same philosophy as the interactive NAUTILUS method [18]. The NAUTILUS method is based on the assumptions that past experiences affect DM's hopes and that people's reactions to gains and losses are asymmetric and, thus, a need for trading off between Pareto optimal solutions during the solution process may hinder the DM from finding desirable solutions. This method starts from a nadir point, a vector whose components are the worst objective function values of solutions and aids the DM to obtain improvement in each objective in the direction specified by her/him progressively without a need of trading off. In every step taken by the DM, information about the part of the Pareto front dominating the current solution (i.e., part which still can be reached without trade-off) is provided. With this information (s)he can re-specify her/his preference information and new solutions are generated in this direction. Although only the last solution will be Pareto optimal (guaranteed by MCDM tools used), a solution is obtained at each step which dominates the previous one and each new Pareto optimal solution is obtained by minimizing an achievement scalarizing function including preferences about desired improvements in objective function values.

In the NAUTILUS method, the starting solution is intentionally bad and the algorithm progressively improves the initial solution to obtain a Pareto optimal solution satisfying the DM, trying to avoid a possible anchoring effect [22] developed by expectations from previous (Pareto optimal) solutions. A typical evolutionary algorithm also has similar traits as that of the NAUTILUS method. In an evolutionary algorithm we start with a random initial population which is usually far from the optimal solutions, similar to the nadir point used in the NAUTILUS method. Additionally, an evolutionary algorithm drives the initial population to the optimal solutions progressively by generating better solutions than the initial population, as NAUTILUS does.

Our PIE algorithm is an "evolutionary algorithm in MCDM" approach. A starting solution is chosen from the initial population based on the preference information of the DM. Subsequently (s)he directs the search deciding from which solution and in which direction the algorithm has to look for solutions dominating the current one. Furthermore, (s)he sets the speed of approaching the Pareto front. The method enables the DM to consider solutions where all objectives can be improved without a need to sacrifice in anything. This should enable a free search towards a desirable part of Pareto optimal solutions. The idea is not to get into a situation where tradeoffs are needed "too early" in the solution process because the need to give up in some objective may hinder the DM from looking for new solutions. As in the NAUTILUS method, although we ensure Pareto optimality only for the final solution (once we have reached the Pareto front), we present solutions that improve the previous ones step by step, what is likely to be attractive for the DM, as people prefer to get better results.

When the DM chooses to select a new solution and direction to orient the search, (s)he is not forced to select it from the presented solutions but can decide to move backwards or present a new point describing her/his aspirations. The PIE algorithm stores all the solutions of the evolutionary algorithm in an archive set to be able to recover previous solutions if the DM desires to go backwards.

The rest of the paper is organized as follows. In Section 2, we introduce some concepts needed for our algorithm. Then, the PIE algorithm is presented in Section 3, where a detailed description of the step-by-step scheme is explained, with some technical issues. Section 4 describes a numerical example demonstrating the method. Finally, conclusions are drawn in Section 5.

2 Concepts

We consider multi-objective optimization problems of the form:

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S, \end{array} \tag{1}$$

where we want to optimize k ($k \geq 2$) conflicting objective functions $f_i : S \rightarrow \mathbb{R}$ simultaneously. Here S is the *feasible set* in the *decision space* $\mathbb{R}^n$ and *decision vectors* are $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in S$. Vectors in the image set of S are *objective vectors* in the *objective space* $\mathbb{R}^k$. A decision vector $\overline{\mathbf{x}} \in S$ is *Pareto optimal* if no objective function can achieve an improvement without a loss in other objective(s), in other words, if there is no other $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\overline{\mathbf{x}})$ for all $i = 1, ..., k$ and $f_j(\mathbf{x}) < f_j(\overline{\mathbf{x}})$ for at least one $j \in 1, ..., k$. The corresponding objective vector is called a *Pareto optimal* objective vector. We say that a vector $\mathbf{z}^1 \in \mathbb{R}^k$ dominates another vector $\mathbf{z}^2 \in \mathbb{R}^k$, denoted by $\mathbf{z}^1 \leq \mathbf{z}^2$, if $z_i^1 \leq z_i^2$ for all $i = 1, ..., k$ and $z_j^1 < z_j^2$ for at least one $j \in 1, ..., k$.

When we want to solve a multi-objective optimization problem, we are interested in finding the Pareto optimal solution that best satisfies the desires of a *decision maker*, a person who is interested in solving the problem and can express preferences in some way. Two important objective vectors are defined as they fix the ranges of the Pareto front, the *ideal objective vector* and the *nadir objective vector*. The components z_i^* of an *ideal objective vector* $\mathbf{z}^* \in \mathbb{R}^k$ are obtained by minimizing each objective function individually, that is, $z_i^* = \min\limits_{\mathbf{x} \in S} f_i(\mathbf{x})$ for all $i = 1, ..., k$. The components $\mathbf{z}_i^{nad}$ of a *nadir objective vector* $\mathbf{z}^{nad} \in \mathbb{R}^k$ are obtained by considering the maximum objective function value for each objective function across all solutions in the Pareto front. The nadir objective vector is generally more difficult to obtain and typically we approximate it [6,16].

The proposed PIE algorithm is based on reference points, introduced in [23]. A *reference point* is an objective vector $\mathbf{q} \in \mathbb{R}^k$ typically provided by the DM. Using a reference point, an *achievement scalarizing function* (ASF) [23] can be formed and optimized to find a solution that best satisfies the aspirations of the DM. By minimizing ASF, we project the reference point $\mathbf{q}$ in the direction specified by a weight vector $\mu = (\mu_1, \mu_2, \ldots, \mu_k)$ to find the closest Pareto optimal solution. For details of ASFs, see [16].

Given a reference point $\mathbf{q} \in \mathbb{R}^k$, an example of an ASF [23] is given by:

$$\begin{aligned} &\text{minimize} \quad \max_{i=1,\dots,k} \{ \mu_i(f_i(\mathbf{x}) - q_i) \} + \rho \sum_{i=1}^{k} (f_i(\mathbf{x}) - q_i) \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{2}$$

where $\rho > 0$ is a so-called augmentation coefficient. Different values for the weights have been proposed in [15]. Among the main advantages of the use of ASFs, we can highlight that the optimal solution of (2) is always a Pareto optimal solution of (1) and that any (properly) Pareto optimal solution of (1) can be obtained by just solving problem (2) with different reference points [16]. Reference points can be of two types: aspiration points (objective function values that are desirable to the DM) and reservation points (objective function values that should be attained, if possible). Additionally, a reference point can be referred to as *attainable*, if the solution obtained by projecting the reference point on to the Pareto front dominates the reference point and otherwise referred to as *unattainable*. In our study, we consider a reference point ($\mathbf{q}$) to be a reservation point.

In [20,24], a different ASF is proposed if both aspiration values q_i^a and reservation values q_i for each objective function are available:

$$\begin{aligned} &\text{minimize} \quad \max_{i=1,\dots,k} \begin{cases} -1 + \alpha \frac{f_i(\mathbf{x}) - q_i^a}{q_i^a - z_i^*} & \text{if } z_i^* \leq f_i(\mathbf{x}) \leq q_i^a \\ \frac{f_i(\mathbf{x}) - q_i}{q_i - q_i^a} & \text{if } q_i^a \leq f_i(\mathbf{x}) \leq q_i \\ \beta \frac{f_i(\mathbf{x}) - q_i}{z_i^{nad} - q_i} & \text{if } q_i \leq f_i(\mathbf{x}) \leq z_i^{nad} \end{cases} \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{3}$$

where α and β are strictly positive numbers. The three cases of expression (3) are defined for three different cases: for attainable reference points in the first case, for unattainable reference points in the third case and the second case is suitable when $\mathbf{q}$ is attainable, but $\mathbf{q}^a$ is not. For more details, see [20]. In our PIE algorithm we can use (2) or (3) depending on the type of preference information available from the DM.

3 The Proposed PIE Algorithm

The PIE algorithm is an interactive preference based evolutionary algorithm, to aid the DM to find a solution satisfying her/his preference information. An evolutionary algorithm is used here because an evolutionary algorithm starts from an initial random population, which is usually far from the Pareto front. Subsequently, the population is driven progressively towards the Pareto front. The population at every generation can be saved in an archive, which not only helps the DM to examine previous solutions, but also to generate a new initial population for an evolutionary algorithm used to solve a new scalarized problem formulated with the new preference information from the DM.

Both the PIE algorithm and an evolutionary algorithm involve seperately a sequence of steps, which are repeated until a termination criterion is satisfied. To make a distinction, we refer to a step of the PIE algorithm as an iteration and a step of an evolutionary algorithm as a generation. Next, we describe the essential components of the PIE algorithm and then present the PIE algorithm.

3.1 Components of the PIE Algorithm

1. **Approximation of nadir point.** To start the PIE algorithm, we need an approximation of the nadir point, which will be the first reference point. We can generate an initial random population and the DM can either decide to select an approximation of the nadir point from a sample of solutions of this population or start from an approximated nadir point constructed by using the worst objective value in every objective function of this initial population. Alternatively, we can apply the approach of [6] or ask the DM to specify worst thinkable objective function values. During the algorithm, the DM is free to choose any other solution as a reference point according to her/his preferences.
2. **Preference information as weights.** Once the DM has chosen a reference point $\mathbf{z}^h$ at iteration h, (s)he has to specify some preference information. As in [15,18], the DM is asked to express her/his preference information and the search for more preferred solutions is oriented accordingly. This preference information can be provided e.g. either by ranking objectives into index sets or by sharing 100 credits among all the objectives.
   (a) **LPF1**: The DM can rank the objectives according to the relative importance of improving each current objective value. (S)he assigns objective functions to classes in an increasing order of importance for improving the corresponding current objective value z_i^h. This importance evaluation allows us to allocate the k objective functions into index sets J_r which represent the importance levels $r = 1, ..., s$, where $1 \leq s \leq k$. If $r < t$, then improving the current objective function values in the index set J_r is less important than improving the current objective function values in J_t. Each objective function can only belong to one index set, but several objectives can be assigned to the same index set J_r.
   The weights for objectives f_i in J_r are defined in [15,18] as follows:

$$\mu_i^h = \frac{1}{r(z_i^{nad} - z_i^{**})}, \ i = 1, ..., k.$$

   (b) **LPF2**: Alternatively, in order to find solutions that reflect the preferences of the DM, the DM can specify percentages expressing how much (s)he would like to improve the current objective values. Then, we ask the following question: *Assuming you have one hundred credits available, how would you distribute them among the current objective values so that the more credits you allocate, the more improvement of the corresponding current objective value is desired?.*

Let us assume that the DM has given p_i^h credits to the objective function f_i. We set $\Delta q_h^i = \frac{p_i^h}{100}$ and then, the weights are formulated in [15]:

$$\mu_i^h = \frac{1}{\Delta q_h^i (z_i^{nad} - z_i^{**})}, \; i = 1, ..., k.$$

We assume that $1 \leq p_i^h \leq 100$, for all $i = 1, ..., k$.

However, the mode of providing preference information may not be limited to the above two ways. Once this preference information is obtained from the DM, we formulate an ASF, which is solved using an evolutionary algorithm.

3. **Aspiration and reservation levels.** Instead of choosing the reference point, the DM can specify bounds on the objective functions as aspiration and reservation levels, $\mathbf{q}^a$ and $\mathbf{q}$ respectively. Here, we replace an ASF with an aspiration-reservation scalarizing function (3). It must be noted that the DM may provide unattainable aspiration levels, in which case, the Pareto optimal solution obtained by solving problem (3) will not satisfy her/his expectations completely. In order to make him/her conscious of that situation, we make the following remark: *The aspiration levels are too optimistic and cannot be achieved.*
4. **Termination criterion for evolutionary algorithm.** As mentioned earlier, we use an evolutionary algorithm to solve the ASF problem and define a termination criterion based on the ASF using the fact that the optimal ASF value is *zero* for Pareto optimal reference points [16,23]. At each η generations, we initialize the reference point used in the ASF with the objective function values of the solution having the smallest ASF value at this generation. Then, the ASF formulated with this reference point is tracked for every $\eta + 10$ generations. If the running average of the ASF values for $\eta + 10$ generations is close to *zero* i.e., 10^{-05}, then the evolutionary algorithm is terminated. This termination criterion ensures that no further improvement in the ASF value is possible or in other words, an individual, $\mathbf{z}^{(h)}$, with the lowest ASF value is in the proximity of the Pareto front.

Next, we describe the step-by-step procedure involved in the PIE algorithm.

3.2 PIE Algorithm

Step 1: Calculate the ideal point $\mathbf{z}^*$ and generate an initial random population P_0 of feasible solutions.

Step 2: The nadir point $\mathbf{z}^{nad}$ is found among the solutions of P_0 depending on the information the DM wishes to give:

(a) If the DM desires to choose $\mathbf{z}^{nad}$ from P_0, we ask: *How many solutions would you like to see to identify an approximation of the nadir point?* and we refer to the number of solutions specified by the DM as N. Next, P_0 is divided into N clusters and one solution from each cluster is presented to the DM. If a cluster has more than one solution, we select the solution closest to the centroid of that cluster. We present these N solutions to the DM, from which (s)he selects one as $\mathbf{z}^{nad}$ (or use ideas of Subsection 3.1).

(b) If the DM is not able to give any information, $\mathbf{z}^{nad}$ is constructed of the worst objective values present in the initial population P_0.

Step 3: Let A and AP be the archive sets of solutions and populations, respectively, defined in Subsection 3.3. Set $A = \varnothing$ and $AP = \varnothing$. Let $\mathbf{z}^1$ be the first reference point and h the iteration number. Set $\mathbf{z}^1 = \mathbf{z}^{nad}$ and $h = 1$.

Step 4: The DM has to provide information about the local improvement of each objective at the current reference point, $\mathbf{z}^h$, in one of the two proposed ways (LPF1 and LPF2) and the weights, μ_i^h are defined accordingly for all $i = 1, \ldots, k$.

Step 5: Set $\mathbf{q} = \mathbf{z}^h$ and $\mu_i = \mu_i^h$ for all $i = 1, \ldots, k$. An achievement scalarizing problem is formulated as in (2). Go to step 7.

Step 6: Formulate the aspiration-reservation scalarizing function (3) with $\mathbf{q}^a = \mathbf{z}^h$ and $\mathbf{q} = \mathbf{z}^{p_{h-1}}$.

Step 7: A single-objective evolutionary algorithm is used to solve the scalarizing function with the termination criterion of Subsection 3.1. Set the initial population needed to start the evolutionary algorithm as explained in Subsection 3.3. At each generation of the evolutionary algorithm, the solution with the lowest ASF value is stored in A and the current population is saved in AP.

Step 8: When the evolutionary algorithm has terminated, select the solution in the final population with the smallest ASF value, $\mathbf{z}^{(h)}$, to be an approximation of the projection of $\mathbf{z}^h$ in the Pareto front. If the solved problem is of the type (3) and $\mathbf{z}^{(h)}$ is dominated by $\mathbf{q}$, the DM is told: *The aspiration levels are too optimistic and cannot be achieved.*

Step 9: Calculate the Euclidean distance d_P between $\mathbf{z}^h$ and $\mathbf{z}^{(h)}$ as $d_P = \sum_{i=1}^{k} \frac{|z_i^{(h)} - z_i^h|}{k|z_i^{nad} - z_i^*|}$, where $|\cdot|$ is the absolute value. Ask the DM: *The distance is d_P between the current reference point $\mathbf{z}^h$ and the corresponding projected Pareto optimal solution $\mathbf{z}^{(h)}$. At what percentage of that distance from $\mathbf{z}^{(h)}$ do you like to investigate the next solution?*. The percentage is referred to as p_h. Next, calculate the point $\mathbf{z}^{p_h}$ in the objective space at p_h of the distance d_P from $\mathbf{z}^{(h)}$.

Step 10: Show current iteration solution. Show $\mathbf{z}^{p_h}$ to the DM. If the DM is satisfied with $\mathbf{z}^{p_h}$, set $\mathbf{z}^h = \mathbf{z}^{p_h}$ and go to step 12, else go to step 11.

Step 11: New reference point. Set $h = h+1$. The DM has to select the next reference point $\mathbf{z}^h$ in one of the following ways:

(a) Otherwise ask the DM: *Would you like to continue the search from* $\mathbf{z}^{p_{h-1}}$*?* If yes, set $\mathbf{z}^h = \mathbf{z}^{p_{h-1}}$. Next, ask the DM: *Would you like to specify new local improvement information at the new reference point,* $\mathbf{z}^h$*?* If yes, go to step 4. Else, go to step 9.

(b) Otherwise ask the DM: *Would you like to examine some solutions in the previous generations?* If yes, we ask: *How many solutions would you like to examine?* and we refer to this number as N. Find N solutions in A as described in Subsection 3.3 and show them to the DM. If the DM finds an interesting solution (set it as $\mathbf{z}^h$) and wishes to examine more solutions around $\mathbf{z}^h$, we ask: *How many solutions would you like to examine around* $\mathbf{z}^h$*?* and we refer to this number as N. Find N solutions in AP as described in Subsection 3.3 and show them to the DM. Finally, when the DM has found a new reference point $\mathbf{z}^h$, ask: *Would you like to specify a new local improvement information at the new reference point,* $\mathbf{z}^h$*?* If yes, go to step 4. Else, set $\mu_i^h = \mu_i^{h-1}$ for all $i = 1, \ldots, k$ and go to step 5.

(c) Otherwise ask the DM: *Would you like to provide an aspiration point?* If yes, set it as $\mathbf{z}^h$ and go to step 6. If no, go to Step 12.

Step 12: Using $\mathbf{z}^h$ as the reference point, solve problem (2) with any suitable mathematical programming technique. The resulting solution is declared as the solution preferred by the DM satisfying her/his preference information.

Step 12 is applied to guarantee at least the local Pareto optimality of the final solution. Furthermore, the solution $\mathbf{z}^h$ is shown to the DM only if the DM sets a 0% distance.

3.3 The Archive Sets A and AP

Archive sets of solutions A and populations AP are maintained during the evolutionary algorithm. At each generation of the evolutionary algorithm, the solution with the ASF value closest to *zero* is stored in A and the corresponding population is saved in AP. The archive sets allow an easy access to previous solutions when the DM decides to move backwards or examine some solutions around the current solution. Additionally, they enable to find an initial population for a new run of the evolutionary algorithm when a new reference point is specified by the DM. Storing many solutions should not be considered as a disadvantage. The high performance and large memory of current computers enable us to store a large number of solutions and to look for some solutions among them. In addition, use of a *structured query language* to handle large number of solutions can be explored.

To show some solutions of previous generations or around $\mathbf{z}^{p_{h-1}}$ in Step 11 (b), N solutions in the archive set A with the smallest Euclidean distances to $\mathbf{z}^{p_{h-1}}$ are found, where N is the number of solutions the DM wants to examine. Additionally, if the DM wants to examine N solutions around any solution $\mathbf{s}^i$, we recover the population $P_{\mathbf{s}^i}$ corresponding $\mathbf{s}^i$ from AP. Next, we find the non-dominated solutions in $P_{\mathbf{s}^i}$ and cluster them into N clusters showing one solution from each cluster to the DM. If a cluster has more than one solution, we select the solution closest to the centroid of that cluster. Here, we present only the non-dominated solutions.

To set the initial population P needed to run the evolutionary algorithm for solving problem (2) or (3), the archive set AP may decrease the computational effort. If $h = 1$, P is set as the initial random population P_0 generated to find the approximation of the nadir point. If $h > 1$, let $\mathbf{s}$ be the point in A with the smallest Euclidean distance to $\mathbf{z}^h$ and let P_h be the population in AP it belongs to. Then, we set $P = P_h$. In this way, we can save on the number of function evaluations, as instead of starting from a random initial population during each evolutionary algorithm's run, we load a population of solutions that have already been evaluated around the current reference point.

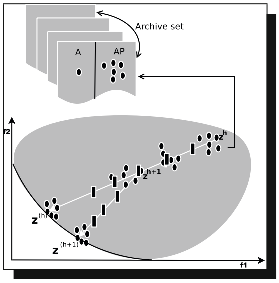


Fig. 1. Graphical idea of the PIE algorithm

A graphical illustration of the solution process in a bi-objective optimization problem is shown in Figure 1, where we start from a reference point $\mathbf{z}^h$ (approximate nadir point). In Figure 1, black circles represent the individuals in the population of an evolutionary algorithm. In every generation, the population in an evolutionary algorithm and the solution with the ASF value close to *zero* is saved in the archives AP and A respectively as shown in Figure 1. As it can be seen, the DM has specified a projection direction from the current reference point towards the Pareto front and the projection $\mathbf{z}^{(h)}$ of $\mathbf{z}^h$ is obtained after running the evolutionary algorithm. Once the DM has examined solutions at different percentages of the distance between $\mathbf{z}^{(h)}$ and $\mathbf{z}^h$, which are shown with black vertical bars on the line joining $\mathbf{z}^h$ and $\mathbf{z}^{(h)}$, the DM has chosen the solution at 50% of the distance from $\mathbf{z}^{(h)}$ as the next reference point $\mathbf{z}^{h+1}$. Next, a new projection direction for $\mathbf{z}^{h+1}$ towards the Pareto front is specified and a new iteration is started to obtain a new solution $\mathbf{z}^{(h+1)}$. Instead of selecting the solution at 50% of the distance from $\mathbf{z}^{(h)}$ as the next reference point, the DM could have examined some solutions in the archives around the solution or provided an aspiration point.

4 Numerical Example

In this section, we illustrate how the PIE algorithm can be used to find the DM's preferred Pareto optimal solution. The selected problem is related to finding the location of a pollution monitoring station [19] and has five conflicting objective functions. The objectives pertain to the expected information loss as estimated by five different experts. The problem is formulated as follows:

$$\begin{aligned}
\text{minimize } & f_1(x_1,x_2) = f(x_1,x_2) \\
& f_2(x_1,x_2) = f(x_1-1.2,x_2-1.5) \\
& f_3(x_1,x_2) = f(x_1+0.3,x_2-3.0) \\
& f_4(x_1,x_2) = f(x_1-1.0,x_2+0.5) \\
& f_5(x_1,x_2) = f(x_1-0.5,x_2-1.7) \\
\text{subject to } & -4.9 \le x_1 \le 3.2 \\
& -3.5 \le x_2 \le 6,
\end{aligned} \tag{4}$$

where

$$f(x_1, x_2) = -3(1 - x_1)^2 e^{-x_1^2-(x_2+1)^2} + 10(\frac{1}{4}x_1 - x_1^3 - x_2^5)e^{-x_1^2-x_2^2}$$
$$-\frac{1}{3}e^{-(x_1+1)^2-x_2^2} + 10.$$

Since the main emphasis of the PIE algorithm is to help the DM in finding a solution satisfying her/his preferences, we have chosen a practical problem with five objectives. It is very common in evolutionary computing literature to demonstrate the efficacy of an algorithm by using test problems. But, the objective functions of these problems have no physical meaning and hence it is very difficult to provide preference information by the DM and to illustrate the working of an interactive algorithm, which is our main goal. The example is used to demonstrate the working steps of the PIE algorithm.

A DM used the PIE algorithm to find an optimal location for a pollution monitoring station. Here, the DM was looking for a location that balances between the five possible losses. For simplicity, we used the preference information to be specified using LPF2 to formulate problem (2). The ideal and nadir vectors as specified in steps 1 and 2 of the PIE algorithm remained unaltered throughout the solution process. A real coded genetic algorithm (RGA) [4] was used as an evolutionary algorithm in the PIE algorithm. However, other evolutionary algorithms can also be used. The parameter used were, a) crossover probability = 0.9, b) mutation probability = 0.5, c) SBX distribution index [4] = 15, and d) mutation distribution index [4] = 20.

To start with the algorithm, we generate an initial random population P_0 and calculate the ideal point $\mathbf{z}^* = (1.87, 1.87, 1.87, 1.87, 1.87)^T$ by individually minimizing each of the objectives subject to constraints. Next we summarize the steps taken by the DM with PIE.

Step 1: Iteration 1: To set the first reference point (an approximation of $\mathbf{z}^{nad}$), the DM decided to investigate *three* solutions of P_0. The solutions provided by the algorithm were $(10.00, 10.00, 10.00, 10.00, 10.00)^T$, $(10.00, 9.88, 4.51, 10.00, 9.51)^T$ and $(9.88, 3.45, 7.37, 9.92, 7.14)^T$ and the DM chose to set the first reference point as $\mathbf{z}^1 = (10.00, 10.00, 10.00, 10.00, 10.00)^T$.

Step 2: The DM specified equal credits to all the objective functions i.e. $\Delta q_1 = (0.2, 0.2, 0.2, 0.2, 0.2)^T$. Using this preference information and $\mathbf{z}^1$, problem (2) was formulated and solved using RGA to get $\mathbf{z}^{(1)}$.

Step 3: The DM was asked to specify the percentage distance, p_1. One at a time, the DM investigated solutions at 80%, 50%, 30% and 20% of the total distance from $\mathbf{z}^{(1)}$.

Step 4: At the solution, $\mathbf{z}^{(20_1)} = (9.17, 9.16, 8.87, 9.14, 9.17)^T$ corresponding to the 20% of the total distance from $\mathbf{z}^{(1)}$, the DM wished to specify the aspiration level, $\mathbf{q}^a = (8, 8, 8, 8, 8)^T$ and we set $\mathbf{q} = \mathbf{z}^{(20_1)}$.

Step 5: Iteration 2: Using the preference information from Step 4, problem (3) was formulated (with $\alpha = 0.1$ and $\beta = 10.0$) and solved using RGA to get $\mathbf{z}^{(2)}$. The algorithm found the aspiration levels in $\mathbf{q}^a = (8, 8, 8, 8, 8)^T$ to be unattainable and subsequently, the DM was told: *The aspiration levels are too optimistic and cannot be achieved.*

Step 6: The DM was asked to specify the percentage distance, p_2. Next, the DM investigated individually solutions at 0%, 50% and 10% of the total distance from $\mathbf{z}^{(2)}$ one after another. Finally, the DM chose the solution at 10% of the distance, $\mathbf{z}^{(10_2)} = (8.98, 8.97, 8.61, 8.95, 8.97)^T$.

Step 7: Iteration 3: Next, the DM decided to examine 10 solutions from A, which were selected as described in Subsection 3.3. After investigating these solutions, the DM found a solution, $(8.95, 8.98, 8.61, 8.90, 8.92)^T$, to be set as the next reference point $\mathbf{z}^3$. Next the DM specified equal credits to all the objective functions i.e. $\Delta q_3 = (0.2, 0.2, 0.2, 0.2, 0.2)^T$.

Step 8: Using this preference information problem (2) was formulated and solved using RGA to get $\mathbf{z}^{(3)}$.

Step 9: The DM was asked to specify the percentage distance, p_3 and the DM investigated individually solutions at 10% and 0% of the total distance from $\mathbf{z}^{(3)}$ one after another.

Step 10: The DM was satisfied with $\mathbf{z}^{(0_3)} = \mathbf{z}^{(3)} = (8.95, 8.98, 8.61, 8.90, 8.94)^T$. Using $\mathbf{z}^{(3)}$ as the reference point, problem (2) was solved using SQP. The resulting Pareto optimal solution $(8.95, 8.98, 8.61, 8.90, 8.94)^T$ was declared as the solution preferred by the DM.

In the above demonstration, the DM learnt during the solution process of what solutions were attainable and did not have to trade off between objectives. He initially chose to give equal credits to all the objective functions as balancing between all five possible cases was equally important. At the end of the first iteration, the DM found a solution $\mathbf{z}^{(20_1)} = (9.17, 9.16, 8.87, 9.14, 9.17)^T$ and then looked for a solution satisfying the aspiration level $\mathbf{q}^a = (8, 8, 8, 8, 8)^T$. After solving problem (3), the algorithm found $\mathbf{z}^{(2)} = (8.96, 8.95, 8.59, 8.93, 8.96)^T$ being dominated by $\mathbf{q}^a$. The DM found a new solution by choosing to look for better solutions in the archive. It can be seen that the new solution $\mathbf{z}^3$ was similar to the previous solution $\mathbf{z}^{(10_2)}$, as there were no better solutions in the archive AP. The solution $\mathbf{z}^3$ was a satisfactory solution to the DM, as the objective function values are all between 8.0 and 9.0 and the DM has struck a balance among the five losses. Note that the solution $\mathbf{z}^{(3)}$ was not improved by SQP, as it was already (locally) Pareto optimal but we cannot know that without solving one more problem. For simplicity, we have carried out a small number of iterations. However, in large scale problems the number of iterations can be higher.

5 Conclusions

In this paper, a new preference based interactive evolutionary algorithm has been proposed. In the proposed algorithm, we start from an approximated nadir point and move progressively towards a desired Pareto optimal solution according to the preferences specified by the DM improving the values of all objectives simultaneously. In this, the DM does not need to compare too many solutions at a time. The ASF defined by the preference information is not only used to obtain better solutions, but also to define an effective termination criterion. At each iteration, we present to the DM just one solution at her/his desired distance from the Pareto front, and move progressively towards the Pareto front, allowing the DM to improve all objectives keeping the cognitive load set to the DM low. Thus, as better solutions are presented gradually, the DM is focussed in the solution process. An evolutionary algorithm has been used to find a near Pareto optimal solution and a mathematical programming technique has been used to guarantee convergence to (locally) Pareto optimal solution. All the populations in the evolutionary algorithm are stored in archive sets and used to show the DM some previous solutions if so desired. Also, we can save in computational cost in the PIE algorithm, if a new run of an evolutionary algorithm is needed. Then, we start from an initial population formed by already evaluated solutions that are located around the new reference point.

We have demonstrated the applicability of the PIE algorithm with a five objective optimization problem. The fact that the objective functions have real meanings enables a realistic interaction with the DM and permits to demonstrate how the PIE algorithm can be used in finding the most interesting solution in few iterations. Next we plan to apply the PIE algorithm with different DMs and problems. Additionally, we are conscious of the importance of a good graphical user-friendly interface in decision making, so a future research direction is to develop a user interface that supports the understandability and easiness of use of the PIE algorithm.

Acknowledgements

The authors would like to thank Professor Francisco Ruiz, Dr. Petri Eskelinen and Dr. Dmitry Podkopaev for their valuable comments.

References

1. Branke, J., Kaubler, T., Schmeck, H.: Guidence in evolutionary multi-objective optimization. Advances in Engineering Software 32(6), 499–507 (2001)
2. Chankong, V., Haimes, Y.Y.: Multiobjective Decision Making Theory and Methodology. Elsevier Science Publishing Co., Inc., New York (1983)
3. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
4. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)
5. Deb, K., Kumar, A.: Light beam search based multi-objective optimization using evolutionary algorithms. In: Proceedings of the Congress on Evolutionary Computation (CEC 2007), pp. 2125–2132. IEEE Press, Los Alamitos (2007)
6. Deb, K., Miettinen, K., Chaudhuri, S.: Towards an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. IEEE Transactions on Evolutionary Computation 14(6), 821–841 (2010)

7. Deb, K., Saxena, D.K.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proceedings of the Congress on Evolutionary Computation (CEC 2006), pp. 3353–3360. IEEE Press, Los Alamitos (2006)
8. Deb, K., Sinha, A., Korhonen, P.J., Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. Tech. Rep. 2009005, KanGAL (2009)
9. Deb, K., Sundar, J., Rao, U.B., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research 2(3), 273–286 (2006)
10. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC 2002), vol. 1, pp. 825–830. IEEE Press, Los Alamitos (2002)
11. Hakanen, J., Miettinen, K., Mäkelä, M.M.: Using genetic algorithms in multiobjective process optimization. In: Bugeda, G., et al. (eds.) Proceedings of the Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2003), CD-Proceedings, CIMNE, Barcelona (2003)
12. Imai, A., Sasaki, K., Nishimura, E., Papadimitriou, S.: Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. European Journal of Operational Research 171(2), 373–389 (2006)
13. Knowles, J., Corne, D.: Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Obayashi, S., et al. (eds.) EMO 2007. LNCS, vol. 4403, pp. 757–771. Springer, Heidelberg (2007)
14. Luque, M., Miettinen, K., Eskelinen, P., Ruiz, F.: Incorporating preference information in interactive reference point methods for multiobjective optimization. Omega 37(2), 450–462 (2009)
15. Luque, M., Ruiz, F., Miettinen, K.: Global formulation for interactive multiobjective optimization. OR Spectrum 33(1), 27–48 (2011)
16. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
17. Miettinen, K.: Using interactive multiobjective optimization in continuous casting of steel. Materials and Manufacturing Processes 22, 585–593 (2007)
18. Miettinen, K., Eskelinen, P., Ruiz, F., Luque, M.: NAUTILUS method: An interactive technique in multiobjective optimization based on the nadir point. European Journal of Operational Research 206(2), 426–434 (2010)
19. Miettinen, K., Lotov, A.V., Kamenev, G.K., Berezkin, V.E.: Integration of two multiobjective optimization methods for nonlinear problems. Optimization Methods and Software 18(1), 63–80 (2003)
20. Ruiz, F., Luque, M., Cabello, J.M.: A classification of the weighting schemes in reference point procedures for multiobjective programming. Journal of the Operations Research Society 60, 544–553 (2009)
21. Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J.: A preference-based evolutionary algorithm for multi-objective optimization. Evolutionary Computation 17(3), 411–436 (2009)
22. Tversky, A., Kahneman, D.: The framing of decisions and the psychology of choice. Science 211, 453–458 (1981)
23. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making: Theory and Application, pp. 468–486. Springer, Hagen (1980)
24. Wierzbicki, A.P.: On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spektrum 8, 73–87 (1986)

Preference Ranking Schemes in Multi-Objective Evolutionary Algorithms

Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck

Institute AIFB, Karlsruhe Institute of Technology
Karlsruhe, D-76128, Germany
{marlon.braun,pradyumn.shukla,hartmut.schmeck}@kit.edu

Abstract. In recent years, multi-objective evolutionary algorithms have diversified their goal from finding an approximation of the complete efficient front of a multi-objective optimization problem, to integration of user preferences. These user preferences can be used to focus on a preferred region of the efficient front. Many such user preferences come from so called proper Pareto-optimality notions. Although, starting with the seminal work of Kuhn and Tucker in 1951, proper Pareto-optimal solutions have been around in the multi-criteria decision making literature, there are (surprisingly) very few studies in the evolutionary domain on this. In this paper, we introduce new ranking schemes of various state-of-the-art multi-objective evolutionary algorithms to focus on a preferred region corresponding to proper Pareto-optimal solutions. The algorithms based on these new ranking schemes are successfully tested on extensive benchmark test problems of varying complexity, with the aim to find the preferred region of the efficient front. This comprehensive study adequately demonstrates the efficiency of the developed multi-objective evolutionary algorithms in finding the complete preferred region for a large class of complex problems.

1 Introduction

Over the past two decades, multi-objective evolutionary algorithms (MOEAs) have been used in finding a well-diverse approximation of the efficient front of a multi-objective optimization problem. In addition to providing the complete front, this approach can also be used to extract innovative design principles [1]. However, information about the complete front burdens the decision-maker and is not especially suitable, if the number of objectives is large. In real-world applications, there is usually one preferred solution which is finally implemented, and it makes sense to use some available information to steer the search towards preferred regions. Hence, we see that various preference based multi-objective evolutionary algorithms have been proposed in the last years [2].

Although the notion of Pareto-optimality has been a central concept in multi-objective optimization, the seminal paper by Kuhn and Tucker in 1951 [3] and later by Geoffrion and others [4, 5], noticed that the use of this notion alone, can give rise to several undesirable properties, like unbounded trade-off's among

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 226–240, 2011.

the objective functions. In order to alleviate these undesirable properties, the notion of proper Pareto-optimality came into existence. Various kinds of proper Pareto-optimal solutions were introduced by researchers and classical methods were devised [6] to find one such solution. Proper Pareto-optimal solutions used additional information (in the form of trade-offs, preference cones, or stability properties of the efficient set [5]) to remove the undesirable Pareto-optimal solutions. Though proper Pareto-optimal solutions have been extensively studied in the classical domain, there is a dearth of studies in the evolutionary domain.

In this paper, we introduce new ranking schemes of existing multi-objective evolutionary algorithms for incorporating preference information. This is done by considering trade-off based preferences that come from various proper Pareto-optimal solutions [4]. We take five state-of-the-art algorithms and propose modifications of their usual ranking schemes. This is used to focus on a preferred region of the efficient front. An extensive computational study on a number of test problems of varying complexity demonstrates the efficiency of new ranking based multi-objective evolutionary algorithms in finding the complete preferred region for a large class of complex problems. Although we only consider preferences coming from the notion of proper Pareto-optimality, the framework that we present is able to handle many kinds of preference structure. This paper adequately demonstrates the niche of population based algorithms in finding the preferred region corresponding to various proper Pareto-optimal solutions.

The paper is structured as follows. The next section presents various notions of proper Pareto-optimality and discusses existing studies. The new ranking schemes of the algorithms are described in Section 3. The fourth section presents extensive simulation results while conclusions as well as extensions which emanated from this study are presented at the end of this contribution.

2 Preliminaries and Existing Studies

Let $f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ and $X \subseteq \mathbb{R}^n$ be given and consider the following multi-objective optimization problem (MOP) and the definition of Pareto-optimality:

$$\min \mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \qquad \text{s.t. } \mathbf{x} \in X.$$

Definition 1 (Pareto-optimality). *A point* $\mathbf{x}^* \in X$ *is called* Pareto-optimal *if no* $\mathbf{x} \in X$ *exists so that* $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ *for all* $i = 1, \ldots, m$ *with strict inequality for at least one index* i.

Let X_p and $\mathcal{E} := F(X_p)$ denote the set of Pareto-optimal solutions and efficient solutions, respectively. A criticism of Pareto-optimality is that it allows unbounded trade-offs. To avoid this, starting with the classical work of Geoffrion [4], various stronger optimality notions have been defined.

Definition 2 (Geoffrion proper Pareto-optimality [4]). *A point* $\mathbf{x}^* \in X$ *is* Geoffrion proper Pareto-optimal *if* $\mathbf{x}^* \in X_p$ *and if there exists a number* $M > 0$ *such that for all* i *and* $\mathbf{x} \in X$ *satisfying* $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$*, there exists an index* j *such that* $f_j(\mathbf{x}^*) < f_j(\mathbf{x})$ *and moreover* $(f_i(\mathbf{x}^*) - f_i(\mathbf{x}))/(f_j(\mathbf{x}) - f_j(\mathbf{x}^*)) \leq M$.

Definition 3 ($\mathcal{M}$-proper Pareto-optimality [7]). *Let a function* $\mathcal{M} : \mathbb{R}^n \to (0, \infty)$ *be given. Then, a point* $\mathbf{x}^* \in X$ *is called* $\mathcal{M}$-proper Pareto-optimal *if* $\mathbf{x}^* \in X_p$ *and if for all* i *and* $\mathbf{x} \in X$ *satisfying* $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$*, there exists an index* j *such that* $f_j(\mathbf{x}^*) < f_j(\mathbf{x})$ *and moreover* $(f_i(\mathbf{x}^*) - f_i(\mathbf{x}))/(f_j(\mathbf{x}) - f_j(\mathbf{x}^*)) \leq \mathcal{M}(\mathbf{x}^*)$.

Although in Definition 3 $\mathcal{M}$ is a function, in this study we use $\mathcal{M} := M$, a constant positive value. This is only for simplicity/ better-illustration and the approach/ ranking-schemes presented in this paper are general enough to handle any function $\mathcal{M}$. Let X_M and $\mathcal{E}_M (:= F(X_M))$ denote the set of M-proper Pareto-optimal solutions and the set of M-proper efficient solutions respectively. It was shown in [7, Lemma 2] that in order to check if a point is (Geoffrion, M-)proper Pareto-optimal or not, it is sufficient to check the boundedness of the trade-offs with the Pareto-optimal points only. This is not evident from Definitions 2 and 3. Based on this, [7] proposed a new ranking scheme in NSGA-II. The algorithm [7] gave a well-diverse representation of $\mathcal{E}_M$ on various test problems. Another work [8] used a constrained approach and it failed for multi-modal problems. In this paper we go on further with the study started in [7] and investigate ranking schemes in various multi-objective evolutionary algorithms.

To the best of our knowledge, we have not found any other work that proposes a method for finding M-proper Pareto-optimal solutions. Note that just the existence of an index $j \in \{1, 2, \ldots, m\}$ in Definitions 2 and 3 is not the same as using a trade-off on fixed objectives (like in [9]) and M-proper Pareto-optimal solutions *cannot* be found by changing the domination. More understanding in this direction is currently under way.

3 Preference Based Ranking Schemes

In this section we take five state-of-the-art algorithms and propose a ranking scheme for these. The first of these is NSGA-II [10], a non-dominated sorting based algorithm (for which we use the same ranking as described in [7]). The other ones are: a steady state version of NSGA-II ssNSGA-II [11], a strength Pareto based algorithm SPEA2 [12], a particle swarm algorithm SMPSO [13], and an algorithm IBEA [14] using indicator based selection. Note that although SMPSO does not use an explicit ranking in the usual sense, it still maintains an archive of non-dominated solutions which are used to guide the search. In this loose way we will call the (new) ranking in SMPSO as the (new) archive update scheme of SMPSO.

Apart from the fact that all of these algorithms are well known and state-of-the-art, they provide an example of the different categories in the gamut of ranking schemes and the way non-dominated solutions are stored/ handled. SPEA2, for example, uses an external archive that stores the non-dominated solutions in a *fixed-size archive*, while SMPSO uses a *variable-size archive* (an upper limit on the archive size always exists) to store the non-dominated solutions. NSGA-II uses explicitly the *non-dominated sorting* to rank the solutions and has no additional archive. IBEA uses a totally different ranking mechanism that is *indicator based*. ssNSGA-II, in contrast to all the other generational algorithms, is a *steady-state* algorithm. The no-free-lunch theorem holds and the aim of this paper is to show how to change the (existing) ranking schemes of various algorithms to guide the search towards proper Pareto-optimal solutions. Hence, we choose the algorithms that have diverse ranking mechanisms and the way non-dominated solutions are used/ stored.

In order to promote those solutions that fulfill the $\mathcal{M}$-proper Pareto optimality condition, we changed the way each algorithm chooses the best solutions. Since every algorithm uses an individual scheme of determining the fitness of a solution, we needed to tailor unique solutions for each algorithm. The general ranking scheme is find the (best) non-dominated set and extract from this the subset of $\mathcal{M}$-proper Pareto solutions. Then, we give this subset a better rank or store them instead of the usual non-dominated set. This is not simple as it seems, although the size of the non-dominated set cannot be zero, the set of $\mathcal{M}$-proper Pareto solutions could be empty. This could cause an (undesirable) empty archive in variable-size archive mechanisms in SMPSO, if we only add the $\mathcal{M}$-proper Pareto solutions instead of the non-dominated ones. Moreover, for other algorithms (NSGA-II, ssNSGA-II, SPEA2, and IBEA) we have to guarantee that the worst solution of the $\mathcal{M}$-proper Pareto set (if this is non-empty) has a better fitness than the best solution of the the non-$\mathcal{M}$-proper Pareto solutions of the non-dominated set. This requirement preserves the Pareto dominance compliant mechanism (see the discussion in [14]).

Next, we detail the new ranking schemes (or fitness assignment in NSGA-II, ssNSGA-II, SPEA2, IBEA and archive update scheme in SMPSO, as we said earlier) of the five algorithms. The name of the new algorithms working using the new ranking schemes are prefixed with 'p', so pIBEA is the IBEA algorithm with the new ranking scheme and so on.

3.1 Ranking in pNSGA-II and pssNSGA-II

NSGA-II and ssNSGA-II use a non-dominated sorting to rank the current population. After this sorting, the new parent population is filled by solutions of different non-dominated fronts, one at a time starting with the first (i.e., the best), second, third fronts and so on. The only difference between NSGA-II and its steady state version ssNSGA-II is the size of the offspring population that is created (in NSGA-II it equals the parent population size while in ssNSGA-II its always one). The new ranking in pNSGA-II (see [7]) and in pssNSGA-II is as follows (where Original ranking denotes the usual ranking in NSGA-II):

1. Do the (usual) non-dominated sorting
2. Take the first front and extract the $\mathcal{M}$-proper Pareto optimal solutions into a new set of solutions called $\mathcal{M}$PPO
3. **If** $\mathcal{M}$PPO and the remaining first front is not empty
   (a) Increase the rank of every front by one
   (b) Assign $\mathcal{M}$PPO the rank 1 and **Return.**
4. **Return:** Original ranking.

Note that in pNSGA-II and pssNSGA-II, and in the following p-algorithms, everything else remains unchanged (e.g., in pNSGA-II and pssNSGA-II, we still do the crowding distance sorting, tournament selection and so on).

3.2 Ranking in pSPEA2

SPEA2 is an evolutionary algorithm having a fixed archive size. It assigns fitness values to the archive and the current population. The new archive is then filled with the solutions having the best (lowest) fitness values. The new ranking in pSPEA2 is as follows:

1. Calculate fitness values according to SPEA2
2. Denote the set of all solutions having a fitness < 1 with P-set (this is the Pareto optimal set)
3. Extract from the P-set the $\mathcal{M}$-proper Pareto optimal solutions into a new set of solutions called $\mathcal{M}$PPO
4. **If** $\mathcal{M}$PPO and the remaining P-set is not empty
   (a) Increase the fitness of all solutions which are not in $\mathcal{M}$PPO by one
   (b) **Return:** Population with changed fitness values.
5. **Return:** Population with original fitness values.

The environment selection mechanism works by first copying all the $\mathcal{M}$PPO solutions, i.e., those which have a fitness less than one, from the archive and the population to, the archive of the next generation and this ensures that the $\mathcal{M}$PPO solutions are promoted.

3.3 Ranking in pSMPSO

SMPSO is also an archive based evolutionary algorithm, but it uses a particle swarm optimization inspired mechanism for searching for Pareto-optimal solutions. In each iteration the algorithm checks which particles of the current generation are to be added to the archive. This is done (here) particle by particle. We changed the adding process by including a $\mathcal{M}$-proper Pareto optimality aggregation of the archive during the adding step. The new ranking (archive update) in pSMPSO is as follows:

1. Denote the solution which is to be added by Solution 1
2. Iterate through the archive and
   (a) Delete all solutions in the archive which are dominated by Solution 1
   (b) **If** Solution 1 is dominated or there is a duplicate of Solution 1, discard it and **Return.**
3. Add Solution 1 to the archive
4. Extract from the archive all solutions which are $\mathcal{M}$-proper Pareto optimal and denote this set as $\mathcal{M}$PPO
5. **If** $\mathcal{M}$PPO is not empty, overwrite the current archive with $\mathcal{M}$PPO
6. **If** the archive size has exceeded its maximum limit, do a crowding distance assignment and delete the solution that has the worst value
7. **Return.**

3.4 Ranking in pIBEA

IBEA combines the offsprings with the current generation and then computes the fitness of the union. It then deletes the solution having the lowest fitness value (in this algorithm better individuals have higher fitness values) and updates the fitness value of the union. This is repeated until the union has the maximum archive size. The union is then the archive for the next iteration. Since $\mathcal{M}$-proper Pareto optimal solutions should always be favored before all other solution, we needed to change the fitness assignment in a way that the worst fitness value of a $\mathcal{M}$-proper Pareto optimal solution is lower (better) than the lowest non-$\mathcal{M}$-proper Pareto optimal solution. The new ranking in pIBEA is as follows:

1. Assign / Update fitness according to IBEA
2. Determine all $\mathcal{M}$-proper Pareto optimal solution of the union
3. Denote the worst fitness value of all $\mathcal{M}$-proper Pareto optimal solutions with $\mathcal{M}_w$ and the best fitness value of all non-$\mathcal{M}$-proper Pareto optimal solution with P_b
4. **If** $\mathcal{M}_w < P_b$: Increase the fitness of every $\mathcal{M}$-proper Pareto optimal solutions by $P_b - \mathcal{M}_w + 1$
5. **Return:** Union.

Table 1. HV. Mean and standard deviation, for continuous problems with $M = 5.0$

	pNSGA-II	pssNSGA-II	pSPEA2	pSMPSO	pIBEA
CTP1	$6.08e-01_{3.0e-04}$	$6.10e-01_{1.2e-04}$	$6.09e-01_{3.7e-04}$	$6.10e-01_{1.2e-04}$	$6.07e-01_{4.0e-04}$
CTP7	$6.34e-01_{1.2e-04}$	$6.34e-01_{2.0e-05}$	$6.34e-01_{1.7e-04}$	$6.34e-01_{5.0e-05}$	$6.33e-01_{2.6e-04}$
DEB2DK_k2	$5.65e-01_{3.0e-04}$	$5.67e-01_{1.3e-05}$	$5.67e-01_{2.6e-05}$	$5.67e-01_{1.3e-04}$	$5.65e-01_{1.2e-04}$
DEB2DK_k4	$5.40e-01_{3.1e-04}$	$5.42e-01_{2.7e-05}$	$5.42e-01_{3.2e-05}$	$5.42e-01_{8.0e-05}$	$5.35e-01_{3.5e-04}$
DEB3DK_k1	$7.43e-01_{3.8e-03}$	$7.47e-01_{2.4e-03}$	$7.51e-01_{2.1e-03}$	$7.50e-01_{1.6e-03}$	$7.50e-01_{1.4e-03}$
DEB3DK_k2	$7.23e-01_{4.1e-03}$	$7.26e-01_{2.9e-03}$	$7.36e-01_{1.1e-03}$	$7.30e-01_{2.7e-03}$	$7.32e-01_{4.5e-04}$
DO2DK_k1	$7.72e-01_{2.8e-04}$	$7.73e-01_{2.1e-05}$	$7.73e-01_{3.5e-05}$	$7.73e-01_{3.3e-05}$	$7.72e-01_{5.9e-04}$
DO2DK_k4	$6.32e-01_{2.2e-04}$	$6.34e-01_{9.9e-06}$	$6.34e-01_{2.3e-05}$	$6.34e-01_{4.0e-04}$	$6.33e-01_{1.2e-04}$
DTLZ2	$3.88e-01_{4.1e-03}$	$3.90e-01_{3.5e-03}$	$3.98e-01_{1.1e-03}$	$3.83e-01_{4.1e-03}$	$3.97e-01_{8.8e-04}$
DTLZ3	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$3.26e-01_{8.9e-02}$	$1.21e-03_{8.5e-03}$
DTLZ4	$3.83e-01_{5.4e-02}$	$3.78e-01_{7.6e-02}$	$3.26e-01_{1.1e-01}$	$3.38e-01_{3.0e-02}$	$2.19e-01_{1.4e-01}$
DTLZ5	$8.71e-02_{5.2e-05}$	$8.75e-02_{4.0e-05}$	$8.73e-02_{9.6e-05}$	$8.75e-02_{6.3e-04}$	$8.67e-02_{6.2e-05}$
DTLZ6	$0.00e+00_{0.0e+00}$	$2.44e-06_{1.7e-05}$	$0.00e+00_{0.0e+00}$	$8.09e-02_{2.4e-02}$	$6.24e-02_{1.4e-02}$
DTLZ7	$2.62e-01_{2.8e-03}$	$2.62e-01_{2.7e-03}$	$2.62e-01_{7.1e-03}$	$2.66e-01_{4.3e-03}$	$2.31e-01_{3.9e-02}$
DTLZ8	$5.70e-01_{1.7e-02}$	$5.68e-01_{1.9e-02}$	$5.51e-01_{1.4e-02}$	$6.14e-01_{1.3e-02}$	$5.42e-01_{1.5e-02}$
DTLZ9	$5.98e-02_{3.5e-03}$	$5.66e-02_{3.2e-03}$	$5.22e-02_{2.8e-03}$	$7.66e-02_{3.8e-03}$	$5.78e-02_{3.6e-03}$
LZ09_F1	$5.73e-01_{1.5e-02}$	$5.64e-01_{1.4e-02}$	$5.40e-01_{1.9e-02}$	$5.98e-01_{2.4e-02}$	$5.41e-01_{1.6e-02}$
LZ09_F4	$5.50e-01_{8.4e-03}$	$5.46e-01_{2.5e-02}$	$5.39e-01_{9.5e-03}$	$5.03e-01_{4.2e-02}$	$5.14e-01_{6.4e-02}$
SZDT1	$6.12e-01_{2.8e-03}$	$6.15e-01_{2.5e-03}$	$6.05e-01_{3.5e-03}$	$6.76e-02_{5.6e-02}$	$6.15e-01_{6.1e-03}$
SZDT2	$2.89e-01_{4.6e-02}$	$2.79e-01_{7.6e-02}$	$1.28e-01_{1.3e-01}$	$0.00e+00_{0.0e+00}$	$2.49e-01_{7.0e-02}$
UF2	$5.51e-01_{3.3e-03}$	$5.50e-01_{2.8e-03}$	$5.47e-01_{3.2e-03}$	$5.36e-01_{4.0e-03}$	$5.48e-01_{3.0e-03}$
UF4	$1.87e-01_{4.3e-03}$	$1.87e-01_{4.3e-03}$	$1.83e-01_{3.3e-03}$	$1.27e-01_{4.4e-03}$	$1.90e-01_{5.8e-03}$
WFG1_2D	$3.87e-01_{9.9e-02}$	$3.24e-01_{1.2e-01}$	$3.04e-01_{9.4e-02}$	$4.31e-02_{4.1e-03}$	$3.02e-01_{1.1e-01}$
WFG1_3D	$5.75e-01_{1.3e-01}$	$5.82e-01_{1.2e-01}$	$4.80e-01_{1.2e-01}$	$5.14e-03_{2.6e-03}$	$3.82e-01_{1.7e-01}$
WFG2_2D	$4.67e-01_{4.5e-04}$	$4.67e-01_{4.8e-04}$	$4.67e-01_{5.0e-04}$	$4.66e-01_{1.5e-03}$	$4.67e-01_{4.6e-04}$
WFG2_3D	$6.75e-01_{6.9e-04}$	$6.76e-01_{5.7e-04}$	$6.75e-01_{8.0e-04}$	$6.76e-01_{4.7e-04}$	$6.74e-01_{9.2e-04}$
ZDT3	$4.69e-01_{3.7e-04}$	$4.70e-01_{2.6e-04}$	$4.68e-01_{9.0e-04}$	$4.65e-01_{2.3e-02}$	$4.68e-01_{6.7e-03}$
ZDT4	$6.18e-01_{7.9e-03}$	$6.16e-01_{3.1e-02}$	$5.84e-01_{3.3e-02}$	$6.31e-01_{2.6e-04}$	$2.70e-01_{9.1e-02}$
ZDT6	$3.78e-01_{3.0e-03}$	$3.93e-01_{1.2e-03}$	$3.54e-01_{7.3e-03}$	$4.01e-01_{9.6e-05}$	$3.86e-01_{2.1e-03}$

Table 2. EPSILON. Median and IQR, for continuous problems with $M = 5.0$

	pNSGA-II	pssNSGA-II	pSPEA2	pSMPSO	pIBEA
CTP1	$6.89e-03_{1.6e-03}$	$4.02e-03_{6.9e-04}$	$6.28e-03_{2.2e-03}$	$4.35e-03_{7.5e-04}$	$8.96e-03_{1.6e-03}$
CTP7	$4.37e-03_{1.1e-03}$	$2.16e-03_{3.3e-04}$	$2.22e-03_{2.6e-03}$	$1.75e-03_{9.8e-05}$	$9.57e-03_{3.0e-03}$
DEB2DK_k2	$1.10e-01_{2.7e-02}$	$4.14e-02_{2.2e-03}$	$5.27e-02_{7.3e-03}$	$4.31e-02_{7.2e-03}$	$1.25e-01_{1.9e-02}$
DEB2DK_k4	$1.06e-01_{3.3e-02}$	$4.12e-02_{2.6e-03}$	$5.40e-02_{6.9e-03}$	$4.74e-02_{1.0e-02}$	$2.69e-01_{1.3e-02}$
DEB3DK_k1	$6.93e-01_{1.6e-01}$	$6.59e-01_{8.5e-02}$	$4.05e-01_{9.6e-02}$	$5.94e-01_{1.2e-01}$	$4.91e-01_{3.7e-02}$
DEB3DK_k2	$7.00e-01_{1.3e-01}$	$6.86e-01_{1.2e-01}$	$4.08e-01_{4.6e-02}$	$5.94e-01_{9.3e-02}$	$5.81e-01_{2.7e-02}$
DO2DK_k1	$2.87e-02_{7.1e-03}$	$1.19e-02_{7.3e-04}$	$1.56e-02_{1.6e-03}$	$1.23e-02_{1.9e-03}$	$2.48e-02_{6.1e-03}$
DO2DK_k4	$2.51e-02_{5.2e-03}$	$9.61e-03_{3.9e-04}$	$1.26e-02_{1.4e-03}$	$9.93e-03_{7.8e-04}$	$1.95e-02_{3.2e-03}$
DTLZ2	$7.96e-02_{2.9e-02}$	$8.11e-02_{3.7e-02}$	$3.63e-02_{1.1e-02}$	$1.08e-01_{1.6e-02}$	$6.94e-02_{1.4e-02}$
DTLZ3	$6.62e+00_{5.1e+00}$	$6.06e+00_{3.6e+00}$	$8.36e+00_{6.5e+00}$	$1.07e-01_{4.9e-01}$	$1.49e+00_{1.5e+00}$
DTLZ4	$6.93e-02_{2.0e-02}$	$6.31e-02_{1.9e-02}$	$4.04e-02_{6.7e-01}$	$1.66e-01_{9.6e-02}$	$7.02e-01_{6.2e-01}$
DTLZ5	$5.29e-03_{1.4e-03}$	$2.62e-03_{4.8e-04}$	$3.11e-03_{6.3e-04}$	$2.80e-03_{3.6e-04}$	$5.81e-03_{8.2e-04}$
DTLZ6	$6.17e-01_{8.4e-02}$	$4.47e-01_{9.3e-02}$	$5.92e-01_{4.9e-02}$	$2.69e-03_{6.1e-04}$	$6.79e-02_{4.0e-02}$
DTLZ7	$5.88e-02_{7.7e-03}$	$5.40e-02_{1.4e-02}$	$4.13e-02_{5.4e-03}$	$6.25e-02_{9.4e-03}$	$1.64e-01_{1.4e+00}$
DTLZ8	$1.34e-01_{2.1e-02}$	$1.33e-01_{1.9e-02}$	$1.60e-01_{1.4e-02}$	$1.00e-01_{3.8e-03}$	$1.51e-01_{1.8e-02}$
DTLZ9	$5.84e-01_{3.4e-02}$	$6.14e-01_{3.0e-02}$	$6.15e-01_{1.7e-02}$	$3.59e-01_{5.6e-02}$	$6.48e-01_{3.0e-02}$
LZ09_F1	$2.06e-01_{6.8e-02}$	$2.41e-01_{4.7e-02}$	$2.80e-01_{5.9e-02}$	$2.47e-02_{1.2e-02}$	$2.96e-01_{3.6e-02}$
LZ09_F4	$2.23e-01_{3.3e-02}$	$2.34e-01_{3.4e-02}$	$2.43e-01_{3.4e-02}$	$2.82e-01_{1.2e-01}$	$2.63e-01_{5.7e-02}$
SZDT1	$1.88e-02_{2.3e-03}$	$1.44e-02_{2.1e-03}$	$2.08e-02_{4.6e-03}$	$6.38e-01_{1.8e-01}$	$1.96e-02_{4.0e-03}$
SZDT2	$4.01e-02_{1.4e-02}$	$3.43e-02_{1.3e-02}$	$7.73e-01_{6.1e-01}$	$2.11e+00_{4.0e-01}$	$2.92e-01_{6.1e-01}$
UF2	$1.36e-01_{1.2e-02}$	$1.38e-01_{1.3e-02}$	$1.41e-01_{1.6e-02}$	$1.98e-01_{3.1e-02}$	$1.34e-01_{1.1e-02}$
UF4	$1.68e-01_{6.2e-03}$	$1.70e-01_{9.3e-03}$	$1.51e-01_{2.6e-02}$	$1.70e-01_{7.7e-03}$	$1.69e-01_{8.6e-03}$
WFG1_2D	$7.33e-01_{4.8e-01}$	$8.02e-01_{4.2e-01}$	$1.07e+00_{2.4e-01}$	$1.22e+00_{4.8e-02}$	$8.01e-01_{2.9e-01}$
WFG1_3D	$3.81e-01_{1.6e-01}$	$3.66e-01_{1.3e-01}$	$6.25e-01_{1.3e-01}$	$1.51e+00_{8.6e-02}$	$5.76e-01_{2.5e-01}$
WFG2_2D	$7.11e-01_{7.1e-01}$	$7.11e-01_{7.1e-01}$	$7.11e-01_{7.1e-01}$	$1.04e-02_{4.2e-03}$	$7.11e-01_{7.1e-01}$
WFG2_3D	$2.34e-01_{2.2e-02}$	$2.36e-01_{1.9e-02}$	$2.11e-01_{1.5e-02}$	$2.59e-01_{2.7e-03}$	$1.86e-01_{3.3e-02}$
ZDT3	$2.21e-03_{3.3e-04}$	$1.38e-03_{2.3e-04}$	$3.79e-03_{1.7e-03}$	$2.64e-03_{3.8e-03}$	$5.30e-03_{2.0e-03}$
ZDT4	$1.79e-02_{9.1e-03}$	$1.06e-02_{6.4e-03}$	$1.45e-01_{1.4e-01}$	$7.61e-03_{1.5e-03}$	$6.50e-01_{1.4e-01}$
ZDT6	$2.50e-02_{6.9e-03}$	$9.22e-03_{1.4e-03}$	$5.34e-02_{1.6e-02}$	$4.80e-03_{3.9e-04}$	$2.01e-02_{3.1e-03}$

4 Experimental Study

In this section, we present an extensive experimental study involving the five algorithms discussed in the previous section (pNSGA-II, pssNSGA-II, pSPEA2, pIBEA, and pSMPSO) and thirty benchmark test problems.

4.1 Experimental Setup

We have written the above p-algorithms in the jMetal framework [15]. The concept of $\mathcal{M}$-proper Pareto optimality is written as a new class `MProperParetoOptimality.java` and is located in the utilities package of the jMetal framework. In its current state, this class may only be used with fixed values for $\mathcal{M}$, however it could easily be modified to use different values or complete functions which may depend on the values of the variables of the corresponding solution and/ or objectives (like $\mathcal{M}$ in Definition 3). The implementation can easily be used in any other multi-objective evolutionary algorithm. During our work with jMetal, we encountered and corrected several errors (e.g., `SMPSO.java` could not work on problems with nonlinear constraints e.g., CTP7). We also implemented new problems into the framework like the CTP and the SZDT suite and the knee problems. The addition of more test suites and the inclusion of proper Pareto-optimal notions builds upon the rich features of jMetal and provides a support for decision making. The source code is made publicly available[1].

The test problems chosen are of varying complexity and are from different test suites that we find in literature. These include two problems from the *CTP suite* (CTP1, CTP7) [16], six from the *Knee suite* (DEB2DK with $k = 2$ and $k = 4$, DEB3DK with $k = 1$ and $k = 2$, DO2DK with $k = 1, s = 0$ and $k = 4, s = 1$,) [17, 18], eight from the *DTLZ suite* (DTLZ2-9, 3 objectives) [19], two from the *LZ09 suite* (LZ09-F1 and LZ09-F4) [20], two from the *CEC-2007 competition* (SZDT1, SZDT2), two from the *CEC-2009 competition* (UF2, UF4), four from the *WFG suite* (WFG1, WFG2, with both 2 and 3 objectives) [21] and four from the *ZDT suite* (ZDT3, ZDT4, ZDT5, ZDT6) [18]. We note that this paper is among the very few studies that consider ZDT5, a difficult discrete problem. The knee and CTP problems included here are also only rarely used in other studies. The inclusion of eight existing test suites makes this study among the most comprehensive experimentation that we find in literature.

For all problems solved, we use a population of size 100 and set the maximum number of function evaluations as 20,000 (200 generations). The archive size (in archive based algorithms) is set to be 100. For continuous problems, we use a standard real-parameter SBX and polynomial mutation operator with $\eta_c = 15$ and $\eta_m = 20$, respectively [18]. Moreover, we use the binary tournament selection in appropriate algorithms. For ZDT5 we use a bit-flip mutation and a single point crossover as recommended in literature.

In this paper, we use $M = 1.5$ and 5.0. Note that these M values restrict the efficient front and this is the preferred efficient front that needs to be found. For all problems we started with a well-distributed approximation of the efficient front (reference set). These fronts were either publicly available (WFGs), analytically constructed (ZDTs) or computed by running NSGA-II with a population size of 5,000 till 2,000 generations, i.e., for 10 million function evaluations (Knee problems). Then, we calculate the preferred points, i.e., the points that satisfy the M-proper Pareto-optimality criteria (from [7, Lemma 2], we only need to consider X_p).

In order to access the quality of our algorithms we used several quality indicators. These indicators describe how well the approximation of the true M-Pareto front is compared to the final M-front returned by each algorithm. Convergence and diversity are two distinct goals of any population based algorithm. We use the additive unary Epsilon indicator and Generational Distance (GD) for measuring the the convergence goal while Spread (for bi-objective problems) and Generalized Spread (for more than two objectives) are used for measuring the diversity goal. Hypervolume (HV) and Inverted Generational Distance (IGD) can measure both convergence and diversity. Except for hypervolume, all the other indicators need to be minimized. For statistical evaluation we run each algorithm for 51 times and present various statistical estimates. This is used together with box-plots for statistical visualization. Moreover, we also use Wilcoxon rank sum

[1] http://www.aifb.kit.edu/web/pNSGA-II/en

test, for making pairwise comparisons between algorithms in order to know about the significance of the obtained results. All of these metrices are computed using the existing jMetal framework.

The data files for all the 51 runs of all the algorithms on all the problems are available on request. This would benefit any other classical and/ or evolutionary comparative studies on proper Pareto-optimal solutions.

4.2 Discussion of Results

In Tables 1, 2, 3 and 4, the entries of the algorithms yielding the best results are highlighted in a dark background, with the second-best is highlighted by a lighter dark background. Table 1 shows the mean and the standard deviation of the HV indicator for the 29 continuous problems. It can be seen that pSMPSO and pssNSGA-II deliver the best results, while pIBEA is the worst. pSMPSO performs exceptionally strongly on the DTLZ family. It also gives very good results for the knee problems, the constrained problems of the CTP and ZDT suit. On the other hand the tables shows, that SMPSO has difficulties solving shifted problems like SZDT, UF or WFG family (these problems were not considered in [13]). Contrary to this pssNSGA-II delivers very good results (with statistical confidence, see Tables 7 and 8) for these difficult problems. The pNSGA-II algorithm also perform quite well here. For smaller values of M (=1.5) the results obtained were similar.

Table 2 shows the Epsilon indicator for $M = 5.0$. Again, we see that pSMPSO and pssNSSGA-II algorithms performs quite well. Although in this study we did not consider the running time explicitly, we found that pSMPSO is the fastest while pssNSGA-II and pIBEA to be the slowest. This is a highly desirable property of pSMPSO. Overalll, pssNSGA-II shows the best results for all different kind of test problems (although the variability can be quite high, see Figure 7). Furthermore its generational counterpart performs also well on the the WFG, LZ and SZDT family. This shows, that the ranking based NSGA-II algorithms keep their niche when preference based selection is introduced. On the other hand, if we look at Tables 3 and 4 we see that slightly different results. pNSGA-II and pSPEA2 perform quite bad in comparison to other algorithms. If we consider the GD indicator, pIBEA shows exceptional results (even for $M = 5$ pIBEA outperforms all other algorithms in terms of GD). An indicator based search performs the best in terms of convergence. This issue opens another interesting direction of research. Instead of using the standard indicators in pIBEA (like HV), can one have a indicator which is based on M-proper Pareto optimality notion itself? This would lead to better focus on finding these solutions.

Looking at Figure 8, we see that the various p-algorithms (with the possible exception of pIBEA) give excellent results on the DTLZ suite. Additionally, the span of pSMPOs boxplots is quite small for nearly all DTLZ problems. This is a very desirable feature, since this means, that pSMPSO provides stable results independent over various runs. We did not test pSMPSO on the discrete problem ZDT5, since its speed constraint mechanism is not designed to handle binary problems. For ZDT5, although the results for IGD and Epsilon may be considered to be quite close, SPEA2 performs slightly better for $M = 1.5$ compared to other algorithms (see Tables 5 and 6). In a sample (random) run plot shown (with $M = 5.0$) in Figure 1 we see that all the algorithms approach the preferred part of ZDT5. The other plots shown in Figures 2-6 show that many of the algorithms do give a well-diverse representation of the preferred region. Especially, the knees in DEB3DK problem (with $k = 2$) are found.

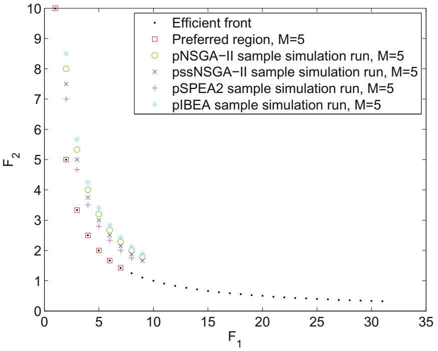


Fig. 1. Preferred front and sample run of the algorithms on ZDT5 ($M = 5.0$)

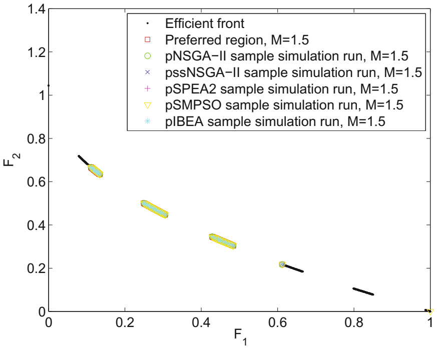


Fig. 2. Preferred front and sample run of the algorithms on CTP7 ($M = 1.5$)

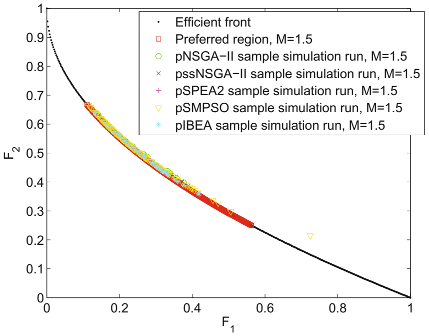


Fig. 3. Preferred front and sample run of the algorithms on LZ1_09F1 ($M = 1.5$)

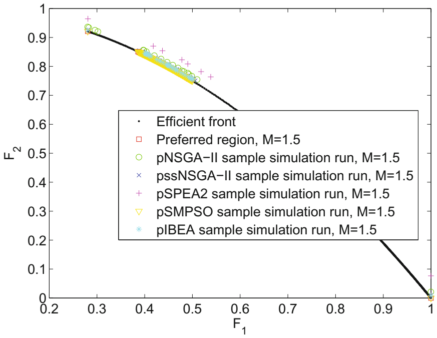


Fig. 4. Preferred front and sample run of the algorithms on ZDT6 ($M = 1.5$)

Table 3. GD. Median and IQR, for continuous problems with $M = 5.0$

	pNSGA-II	pssNSGA-II	pSPEA2	pSMPSO	pIBEA
CTP1	$1.42e-04_{5.8e-05}$	$9.09e-05_{5.8e-05}$	$1.37e-04_{9.0e-05}$	$1.29e-04_{1.3e-04}$	$6.29e-05_{2.6e-05}$
CTP7	$1.17e-05_{9.8e-06}$	$1.06e-05_{5.8e-06}$	$1.69e-05_{2.3e-05}$	$1.21e-05_{3.5e-06}$	$1.05e-05_{1.8e-05}$
DEB2DK_k2	$3.53e-05_{3.2e-06}$	$3.66e-05_{3.4e-06}$	$3.61e-05_{3.9e-06}$	$4.10e-05_{9.3e-06}$	$3.45e-05_{3.5e-06}$
DEB2DK_k4	$3.43e-05_{5.1e-06}$	$3.38e-05_{4.0e-06}$	$3.37e-05_{3.1e-06}$	$5.33e-05_{2.9e-05}$	$3.33e-05_{4.3e-06}$
DEB3DK_k1	$3.71e-03_{9.6e-04}$	$3.37e-03_{6.9e-04}$	$2.69e-03_{6.8e-04}$	$2.67e-03_{8.2e-04}$	$4.75e-03_{7.7e-04}$
DEB3DK_k2	$1.63e-03_{3.4e-04}$	$1.44e-03_{2.6e-04}$	$1.59e-03_{2.4e-04}$	$3.49e-03_{1.0e-03}$	$1.01e-03_{1.0e-04}$
DO2DK_k1	$1.38e-04_{1.0e-05}$	$1.45e-04_{1.1e-05}$	$1.38e-04_{1.4e-05}$	$1.00e-03_{1.4e-02}$	$1.37e-04_{8.2e-06}$
DO2DK_k4	$9.68e-05_{1.3e-05}$	$9.78e-05_{9.1e-06}$	$9.77e-05_{1.0e-05}$	$1.10e-04_{7.6e-03}$	$1.05e-04_{1.3e-05}$
DTLZ2	$6.25e-03_{3.5e-03}$	$8.22e-03_{2.6e-03}$	$5.35e-03_{1.9e-03}$	$1.44e-02_{1.4e-03}$	$3.85e-03_{1.6e-03}$
DTLZ3	$2.57e+00_{2.0e+00}$	$2.21e+00_{1.3e+00}$	$3.06e+00_{2.9e+00}$	$1.37e-02_{8.3e-03}$	$3.86e-01_{5.9e-01}$
DTLZ4	$3.27e-03_{3.4e-03}$	$6.31e-03_{3.6e-03}$	$2.87e-03_{3.7e-03}$	$1.25e-03_{1.2e-03}$	$1.19e-03_{5.3e-04}$
DTLZ5	$2.20e-04_{1.8e-05}$	$2.15e-04_{1.9e-05}$	$2.14e-04_{1.4e-05}$	$2.34e-04_{1.1e-03}$	$2.61e-04_{1.4e-05}$
DTLZ6	$1.29e-01_{1.8e-02}$	$8.46e-02_{1.2e-02}$	$1.05e-01_{1.3e-02}$	$2.27e-04_{1.4e-03}$	$6.28e-03_{7.6e-03}$
DTLZ7	$1.54e-03_{2.2e-04}$	$1.44e-03_{1.7e-04}$	$1.43e-03_{1.6e-04}$	$2.28e-03_{1.2e-03}$	$1.18e-03_{4.5e-04}$
DTLZ8	$9.37e-04_{1.5e-04}$	$8.06e-04_{1.6e-04}$	$8.35e-04_{3.2e-04}$	$2.65e-03_{2.0e-03}$	$3.34e-04_{3.1e-05}$
DTLZ9	$5.44e-04_{2.6e-04}$	$4.13e-04_{1.2e-04}$	$5.96e-04_{2.6e-04}$	$5.34e-03_{2.0e-03}$	$1.68e-04_{1.2e-04}$
LZ09_F1	$1.20e-03_{3.5e-04}$	$1.01e-03_{2.2e-04}$	$1.27e-03_{4.0e-04}$	$1.85e-03_{9.0e-04}$	$6.74e-04_{2.5e-04}$
LZ09_F4	$1.81e-03_{5.5e-04}$	$1.75e-03_{6.3e-04}$	$2.03e-03_{5.3e-04}$	$5.43e-03_{3.2e-03}$	$1.20e-03_{5.3e-04}$
SZDT1	$1.28e-03_{2.0e-04}$	$1.16e-03_{2.7e-04}$	$1.81e-03_{3.4e-04}$	$8.49e-02_{2.9e-02}$	$8.29e-04_{2.1e-04}$
SZDT2	$1.92e-03_{4.7e-04}$	$1.62e-03_{4.2e-04}$	$6.38e-03_{1.6e-01}$	$6.47e-01_{6.0e-01}$	$1.50e-03_{3.9e-04}$
UF2	$2.64e-03_{7.9e-04}$	$2.54e-03_{8.3e-04}$	$2.53e-03_{5.3e-04}$	$5.47e-03_{2.0e-03}$	$1.84e-03_{4.5e-04}$
UF4	$1.38e-02_{5.3e-03}$	$1.47e-02_{3.8e-03}$	$1.70e-02_{3.6e-03}$	$2.40e-02_{2.3e-03}$	$1.23e-02_{7.3e-03}$
WFG1_2D	$7.23e-03_{1.4e-02}$	$1.34e-02_{2.2e-02}$	$7.32e-03_{1.4e-02}$	$8.50e-02_{1.1e-02}$	$1.39e-02_{1.6e-02}$
WFG1_3D	$2.59e-03_{9.0e-03}$	$2.76e-03_{9.1e-03}$	$2.22e-03_{1.2e-02}$	$1.09e-01_{6.3e-03}$	$1.86e-02_{2.1e-02}$
WFG2_2D	$2.28e-04_{7.5e-04}$	$2.17e-04_{7.2e-04}$	$2.14e-04_{7.3e-04}$	$4.50e-03_{4.3e-03}$	$1.96e-04_{7.8e-04}$
WFG2_3D	$1.93e-02_{3.5e-04}$	$1.96e-02_{2.8e-04}$	$1.73e-02_{4.5e-04}$	$1.78e-02_{4.6e-04}$	$1.63e-02_{3.5e-04}$
ZDT3	$8.64e-05_{1.2e-05}$	$7.64e-05_{1.1e-05}$	$1.50e-04_{5.0e-05}$	$3.15e-04_{7.3e-04}$	$6.87e-05_{5.7e-06}$
ZDT4	$7.26e-04_{4.5e-04}$	$5.04e-04_{4.0e-04}$	$9.99e-04_{1.0e-03}$	$1.00e-04_{1.0e-03}$	$3.79e-04_{2.7e-04}$
ZDT6	$1.70e-03_{3.8e-04}$	$6.44e-04_{1.1e-04}$	$3.36e-03_{7.8e-04}$	$9.28e-05_{1.3e-05}$	$7.90e-04_{2.1e-04}$

Table 4. EPSILON. Median and IQR, for continuous problems with $M = 1.5$

	pNSGA-II	pssNSGA-II	pSPEA2	pSMPSO	pIBEA
CTP1	$4.39e-03_{1.2e-03}$	$1.73e-03_{9.6e-05}$	$2.35e-03_{2.7e-04}$	$1.93e-03_{3.2e-04}$	$2.93e-03_{3.4e-04}$
CTP7	$2.11e-03_{4.3e-04}$	$8.83e-04_{6.4e-05}$	$9.89e-04_{1.2e-04}$	$8.61e-04_{8.6e-05}$	$4.37e-03_{8.3e-04}$
DEB2DK_k2	$7.82e-03_{2.3e-03}$	$3.15e-03_{1.6e-04}$	$3.75e-03_{6.5e-04}$	$3.14e-03_{2.5e-04}$	$8.47e-03_{1.1e-03}$
DEB2DK_k4	$2.11e-02_{5.5e-03}$	$7.56e-03_{7.0e-04}$	$8.36e-03_{1.3e-03}$	$7.02e-03_{7.4e-04}$	$4.95e-02_{1.3e-03}$
DEB3DK_k1	$4.31e-02_{3.2e-02}$	$3.10e-02_{1.9e-02}$	$4.10e-02_{3.1e-02}$	$5.95e-02_{1.9e-02}$	$2.97e-02_{1.0e-02}$
DEB3DK_k2	$8.87e-02_{6.0e-02}$	$7.31e-02_{4.8e-02}$	$1.22e-01_{9.4e-02}$	$9.53e-02_{5.6e-02}$	$7.49e-02_{4.9e-02}$
DO2DK_k1	$4.12e-03_{1.5e-03}$	$1.79e-03_{1.0e-04}$	$2.09e-03_{2.4e-04}$	$1.76e-03_{1.2e-04}$	$2.89e-03_{7.0e-04}$
DO2DK_k4	$1.05e-03_{3.9e-04}$	$4.23e-04_{3.6e-05}$	$4.57e-04_{8.9e-05}$	$4.19e-04_{3.9e-05}$	$9.74e-04_{2.9e-04}$
DTLZ2	$4.46e-04_{2.5e-04}$	$3.21e-04_{1.5e-04}$	$2.63e-03_{1.3e-03}$	$1.25e-04_{1.8e-05}$	$1.35e-04_{9.7e-06}$
DTLZ3	$8.31e+00_{6.9e+00}$	$6.80e+00_{4.6e+00}$	$9.65e+00_{4.9e+00}$	$7.07e-01_{9.1e-01}$	$1.03e+00_{1.1e+00}$
DTLZ4	$3.08e-04_{1.5e-04}$	$2.36e-04_{7.5e-05}$	$2.90e-03_{9.8e-01}$	$1.23e-04_{1.8e-07}$	$9.84e-01_{9.8e-01}$
DTLZ5	$2.89e-05_{2.2e-05}$	$1.23e-05_{6.9e-06}$	$4.29e-04_{2.5e-04}$	$4.93e-06_{0.0e+00}$	$5.99e-06_{8.1e-07}$
DTLZ6	$1.29e+00_{1.9e-01}$	$9.68e-01_{2.0e-01}$	$1.34e+00_{1.4e-01}$	$7.05e-01_{1.0e+00}$	$2.25e-02_{4.3e-02}$
DTLZ7	$2.39e-02_{9.3e-03}$	$1.57e-02_{5.3e-03}$	$8.24e-02_{3.0e-02}$	$8.32e-03_{1.5e-03}$	$1.28e+00_{1.3e+00}$
DTLZ8	$2.64e-03_{2.7e-03}$	$2.91e-03_{4.1e-03}$	$6.01e-03_{7.7e-03}$	$1.81e-03_{2.3e-03}$	$9.64e-04_{1.5e-03}$
DTLZ9	$5.72e-01_{3.2e-02}$	$5.97e-01_{2.8e-02}$	$6.33e-01_{3.3e-02}$	$4.62e-01_{1.5e-01}$	$6.60e-01_{3.8e-02}$
LZ09_F1	$1.17e-01_{7.2e-02}$	$1.29e-01_{5.8e-02}$	$1.58e-01_{4.3e-02}$	$6.24e-02_{8.7e-02}$	$1.13e-01_{5.6e-02}$
LZ09_F4	$1.26e-01_{4.6e-02}$	$1.27e-01_{3.9e-02}$	$1.44e-01_{7.8e-02}$	$3.56e-01_{2.4e-01}$	$1.03e-01_{2.6e-02}$
SZDT1	$2.77e-02_{1.4e-02}$	$2.50e-02_{9.6e-03}$	$4.17e-02_{2.3e-02}$	$5.65e-01_{3.0e-01}$	$2.86e-02_{1.6e-02}$
SZDT2	$4.52e-02_{1.7e-02}$	$3.76e-02_{1.2e-02}$	$3.26e-01_{9.6e-01}$	$2.28e+00_{3.9e-01}$	$3.45e-02_{2.2e-01}$
UF2	$2.05e-01_{4.0e-02}$	$1.97e-01_{5.6e-02}$	$2.01e-01_{2.0e-02}$	$2.15e-01_{1.7e-02}$	$1.46e-01_{1.0e-02}$
UF4	$2.02e-02_{1.7e-02}$	$1.90e-02_{1.5e-02}$	$5.01e-02_{2.8e-02}$	$8.23e-02_{2.8e-02}$	$7.72e-03_{1.5e-02}$
WFG1_2D	$5.18e-01_{1.8e-01}$	$6.42e-01_{3.7e-01}$	$6.55e-01_{2.9e-01}$	$1.90e+00_{4.4e-01}$	$8.43e-01_{2.9e-01}$
WFG1_3D	$1.76e-01_{8.9e-02}$	$1.58e-01_{1.8e-01}$	$2.74e-01_{1.9e-01}$	$1.07e+00_{1.4e-01}$	$3.40e-02_{2.1e-01}$
WFG2_2D	$7.12e-01_{6.9e-01}$	$7.12e-01_{6.9e-01}$	$7.12e-01_{6.9e-01}$	$2.55e-02_{2.6e-03}$	$7.12e-01_{6.9e-01}$
WFG2_3D	$2.52e-02_{1.2e-02}$	$2.31e-02_{8.5e-03}$	$3.07e-02_{1.5e-02}$	$2.60e-02_{1.2e-02}$	$2.40e-02_{1.3e-02}$
ZDT3	$1.84e-03_{8.1e-04}$	$9.31e-04_{5.6e-04}$	$5.42e-03_{1.7e-03}$	$2.24e-01_{2.5e-01}$	$3.97e-04_{2.6e-04}$
ZDT4	$1.92e-02_{1.5e-02}$	$1.80e-02_{1.9e-02}$	$3.36e-02_{2.8e-02}$	$3.32e-03_{1.1e-03}$	$1.69e-01_{1.8e-01}$
ZDT6	$2.66e-02_{7.9e-03}$	$1.17e-02_{6.8e-03}$	$6.15e-02_{1.4e-02}$	$6.18e-04_{6.2e-05}$	$1.19e-02_{7.0e-03}$

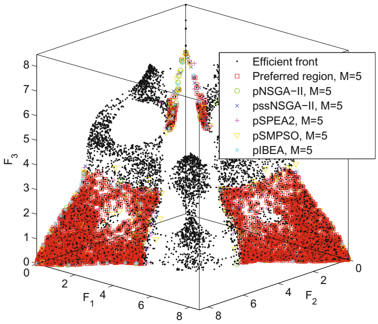


Fig. 5. Preferred front and sample run of the algorithms on DEB3DK_k2 ($M = 5.0$)

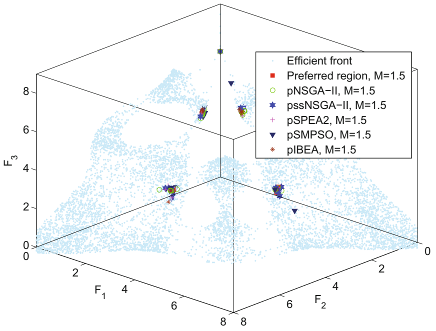


Fig. 6. Preferred front and sample run of the algorithms on DEB3DK_k2 ($M = 1.5$)

Table 5. IGD. Median and IQR, for ZDT5 with $M = 5.0$ and $M = 1.5$

	pNSGA-II	pssNSGAII	pSPEA2	pIBEA
ZDT5, $M = 5.0$	$6.95e-02_{4.6e-03}$	$6.73e-02_{1.5e-03}$	$6.95e-02_{5.1e-03}$	$6.95e-02_{4.6e-03}$
ZDT5, $M = 1.5$	$8.08e-01_{7.1e-02}$	$9.15e-01_{2.4e-01}$	$7.36e-01_{1.3e-01}$	$7.36e-01_{1.3e-01}$

Table 6. EPSILON. Mean and standard deviation, for ZDT5 with $M = 5.0$ and $M = 1.5$

	NSGAII	ssNSGAII	SPEA2	IBEA
ZDT5, $M = 5.0$	$1.01e+00_{3.2e-02}$	$1.01e+00_{3.9e-02}$	$1.01e+00_{3.2e-02}$	$1.00e+00_{0.0e+00}$
ZDT5, $M = 1.5$	$9.90e-01_{6.9e-02}$	$1.03e+00_{7.7e-02}$	$9.48e-01_{1.5e-01}$	$9.74e-01_{1.1e-01}$

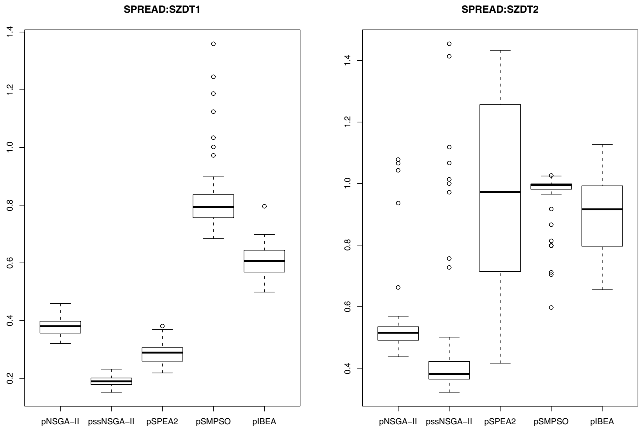


Fig. 7. Boxplots of Spread indicator on SZDT problems

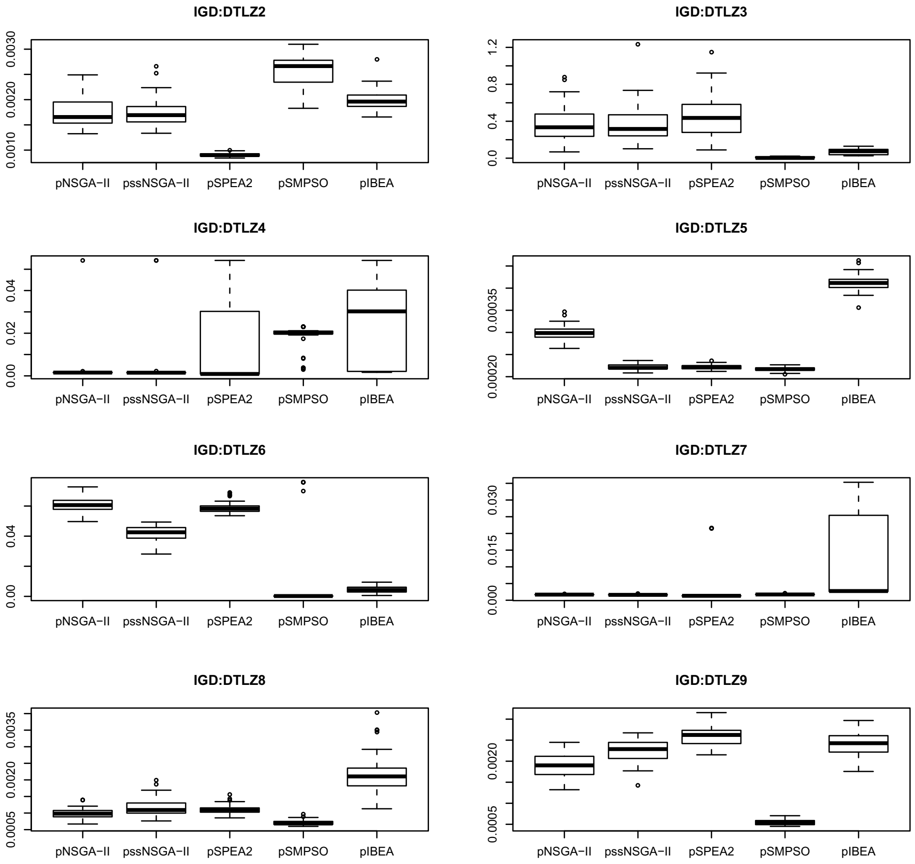


Fig. 8. Box-plots of the results obtained by all the algorithms on DTLZ2-9 problems, with $M = 5.0$

Table 7. Wilcoxon test on SZDT1 and SZDT2 wrt. Spread (left) and EPSILON (right) with M=5.0.

	ssNSGAII	SPEA2	SMPSO	IBEA
NSGAII	▽ ▽	▽ ▲	▲ ▲	▲ ▲
ssNSGAII		▲ ▲	▲ ▲	▲ ▲
SPEA2			▲ –	▲ –
SMPSO				▽ ▽

	ssNSGAII	SPEA2	SMPSO	IBEA
NSGAII	▽ ▽	▲ ▲	▲ ▲	▲ ▲
ssNSGAII		▲ ▲	▲ ▲	▲ ▲
SPEA2			▲ ▲	▽ ▽
SMPSO				▽ ▽

Table 8. Wilcoxon test on WFG1_2D WFG1_3D WFG2_2D WFG2_3D wrt. HV with $M = 5.0$.

	ssNSGAII	SPEA2	SMPSO	IBEA
NSGAII	▲ – – ▽	▲ ▲ – –	▲ ▲ ▲ ▽	▲ ▲ – ▲
ssNSGAII		– ▲ – ▲	▲ ▲ ▲ ▲	– ▲ – ▲
SPEA2			▲ ▲ ▲ ▽	– ▲ – ▲
SMPSO				▽ ▽ ▽ ▲

5 Conclusions

This study proposes new ranking schemes in various state-of-that-art multi-objective evolutionary algorithms to incorporate user preferences that come from proper Pareto-optimality notions. The algorithms based on these new ranking schemes are tested on extensive benchmark test problems of varying complexity with the aim to find the preferred region of the efficient front. Many of these algorithms give a well-diverse representation of M-proper Pareto-optimal solutions. The extensive computational involving thirty problems and 9 test suites shows that the ranking schemes proposed in this paper are not limited to continuous and unconstrained/ bound-constrained problems and they also well on problems having complicated Pareto fronts (like LZ problems) as well as complicated efficient fronts (like DEB3DK problems).

On an another note this study demonstrates the usefulness of considering various test-problem suites. pSMPSO, for example was found not to be good on SZDT, WFG and UF problems, while it one of the best algorithms for DTLZ, ZDT and knee problems. This might be due to the fact that all of these (DTLZ, ZDT and knee) problems have simple Pareto-optimal fronts, although the problems itself are quite difficult.

In this paper we have suggested how multi-objective evolutionary algorithms can be of help to the multi-criteria decision making community. It would be interesting to see how other notions of proper Pareto-optimality, based on cones or on stability properties [5] can be fitted into these or similar ranking schemes. Moreover, emphasis must be placed on developing interactive evolutionary multi-objective optimization techniques for many objective problems, that incorporate the numerous notions of proper Pareto-optimality [6].

References

[1] Deb, K.: Innovization: Extracting innovative solution principles through multiobjective optimization. Springer, Heidelberg (2010)

[2] 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)

[3] Kuhn, H.W., Tucker, A.W.: In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, vol. 1950, pp. 481–492 (1951)

[4] Geoffrion, A.M.: Proper efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications 22, 618–630 (1968)

[5] Makarov, E.K., Rachkovski, N.N.: Unified representation of proper efficiency by means of dilating cones. J. Optim. Theory Appl. 101, 141–165 (1999)

[6] Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)

[7] Shukla, P.K., Hirsch, C., Schmeck, H.: A framework for incorporating trade-off information using multi-objective evolutionary algorithms. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 131–140. Springer, Heidelberg (2010)

[8] Shukla, P.K.: In search of proper pareto-optimal solutions using multi-objective evolutionary algorithms. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4490, pp. 1013–1020. Springer, Heidelberg (2007)

[9] Branke, J., Deb, K.: Integrating User Preferences into Evolutionary Multi-Objective Optimization. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, pp. 461–477. Springer, Heidelberg (2005)

[10] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)

[11] Durillo, J.J., Nebro, A.J., Luna, F., Alba, E.: On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 183–197. Springer, Heidelberg (2009)

[12] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailou, P., Fogarty, T. (eds.) EUROGEN 2001, Athens, Greece (2001)

[13] Nebro, A., Durillo, J., García-Nieto, J., Coello Coello, C., Luna, F., Alba, E.: SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In: MCDM 2009, pp. 66–73. IEEE Press, Los Alamitos (2009)

[14] Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)

[15] Durillo, J., Nebro, A., Alba, E.: The jmetal framework for multi-objective optimization: Design and architecture. In: CEC 2010, Barcelona, pp. 4138–4325 (2010)

[16] Deb, K., Pratap, A., Meyarivan, T.: Constrained test problems for multi-objective evolutionary optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 284–298. Springer, Heidelberg (2001)

[17] Branke, J., Deb, K., Dierolf, H., Osswald, M.: Finding knees in multi-objective optimization. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 722–731. Springer, Heidelberg (2004)

[18] Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)

[19] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In: Freitas, A.A., et al. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer, London (2005)

[20] Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. Trans. Evol. Comp. 13, 284–302 (2009)

[21] Huband, S., Hingston, P., Barone, L., White, L.: A review of multi-objective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10, 280–294 (2005)

Interactive Multiobjective Mixed-Integer Optimization Using Dominance-Based Rough Set Approach

Salvatore Greco[1], Benedetto Matarazzo[1], and Roman Słowiński[2]

[1] Faculty of Economics, University of Catania, 95129 Catania, Italy
{salgreco,matarazz}@unict.it
[2] Institute of Computing Science, Poznań University of Technology, Poznań, and
Systems Research Institute, Polish Academy of Sciences, 00-441 Warsaw, Poland
roman.slowinski@cs.put.poznan.pl

Abstract. We present a new methodology for dealing with interactive multiobjective optimization in case of mixed-integer variables. The preference information elicited by the Decision Maker (DM) in course of the interaction is processed using the Dominance-based Rough Set Approach (DRSA). This permits to ask the DM simple questions and obtain in return a decision model expressed in terms of easily understandable "if..., then..." decision rules. In each iteration, the current set of decision rules is presented to the DM with the proposal of selecting one of them considered the most representative. The selected decision rule specifies some minimal requirements that the DM desires to be achieved by the objective functions. This information is translated into a set of constraints which are added to the original problem restricting the space of feasible solutions. Moreover, we introduce one simple but effective algorithm, called bound-and-cut, that efficiently reduces the set of feasible values of the integer variables. This process continues iteratively until the part of the Pareto front that is interesting for the DM can be exhaustively explored with respect to the integer variables. The bound-and-cut algorithm can be embedded in an Evolutionary Multiobjective Optimization (EMO) method, which permits to compute a reasonable approximation of the considered part of the Pareto front. A subset of representative solutions can be selected from this approximation and presented to the DM in the dialogue phase of each iteration.

1 Introduction

Mixed-integer optimization is a classical problem in operational research. Several methodologies have been proposed to deal with it, both exact and heuristic, such as branch-and-bound [9], branch-and-cut [6], branch-and-price [2], as well as various (meta)heuristics. This problem becomes much more complex in presence of multiple objective functions, i.e. in case of Multiobjecive Optimization (for an updated survey see [3]). Some methods proposed for multiobjective optimization are simple extensions of the methodologies used for single-objective optimization

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 241–253, 2011.

(for a survey see [5]). Very often optimization problems with multiple objective functions involve an interactive methodology, which alternates phases of dialogue with the Decision Maker (DM) with phases of computation (for an updated survey see [3]). When some variable are integer, we speak of Interactive Multiobjective Mixed-Integer Optimization (IMMIO). A certain number of methods has been proposed to deal with IMMIO problems (for an updated survey see [1]).

In this paper we propose a new methodology for IMMIO problems. Assuming, without loss of generality, that all objective functions are to be maximized, our methodology is based on the idea of using some lower bounds on the objective functions, resulting from a decision model representing the preference of the DM. These bounds have a double role: they gradually constrain the feasible set of solutions ensuring convergence to the most preferred solution on the Pareto front, and they reduce the range of variation of the integer variables until the most interesting part of the Pareto front can be exhaustively explored by the DM with respect to feasible values of the integer variables. Of course, in case of minimization of objective functions, we use upper bounds playing the same role.

The method is called *bound-and-cut*, meaning that the lower bounds on objective functions permit to cut-off some range of variation of the integer variables. Observe that bound-and-cut is different from branch-and-cut in which a cut is made by some hyperplanes built on the basis of a non-integer solution of the regular simplex method, such that it cuts-off the current non-integer solution, and does not cut-off any feasible solution with integer values of variables supposed to be integer. Instead, bound-and-cut bounds the range of integer variables by max-min analysis.

The bound-and-cut algorithm can also be applied to single objective optimization problems. It can also be effectively embedded into some Evolutionary Multiobjective Optimization (EMO) methodology [4]. In fact, bound-and-cut algorithm benefits from EMO applied to the computation of lower bounds on objective functions, and, reciprocally, EMO methodology benefits from the bound-and-cut algorithm which reduces the search space permitting to improve the performance of EMO from iteration to iteration. In this context, it is also important to join the two methods through an interactive procedure using a proper interface to the bound-and-cut algorithm. Based on our experience, we advocate for the use of the Dominance-based Rough Set Approach (IMO-DRSA) [8], which provides a convenient interface for this purpose. In fact, apart from its many strong points in terms of transparency and comprehensibility, IMO-DRSA makes use of a representation of the DM's preferences in terms of decision rules whose syntax is: "if objective function f_{j1} takes at least value b_{j_1}, and ..., objective function f_{j_p} takes at least value b_{jp}, then the considered solution is good". Remark that the values $b_{j_1}, \ldots, b_{j_p}$ can be interpreted as the lower bounds on the objective functions required by the bound-and-cut algorithm.

The paper is organized as follows. In the second section, the IMO-DRSA methodology is presented. In the third section, the bound-and-cut algorithm is introduced. In the fourth section, a didactic example illustrates the proposed methodology. The last section contains conclusions.

2 Interactive Multiobjective Optimization Guided by Dominance-Based Rough Set Approach (IMO-DRSA)

Rough Set Theory proposed by Pawlak [10] has proved to be an excellent mathematical tool for the analysis of a vague description of objects, and can be considered as a mathematical basis for structuring data before mining them. However, rough set theory cannot deal with decision problems, because it does not take into account the preference order on evaluations of alternatives with respect to considered criteria. Greco, Matarazzo and Slowinski [7] proposed a generalization of rough set theory called Dominance-based Rough Set Approach (DRSA), which is able to take preference order into account and, therefore, constitutes a very interesting methodology for multiple criteria decision analysis. DRSA accepts as input a set of exemplary decisions (for example classification of alternatives into "good" and "others") and gives in return a preference model in terms of easily understandable "if..., then..." decision rules explaining the exemplary decisions (see [11,12] for tutorial reviews).

Representation of preferences in terms of decision rules induced through Dominance-based Rough Set Approach (DRSA) can be fruitfully exploited in course of an Interactive Multiobjective Optimization (IMO) procedure, as proposed in [8]. An interactive procedure is composed of two alternating stages: computation stage and dialogue stage. In the computation stage, a subset of feasible solutions is calculated and presented to the Decision Maker (DM) (also called user). Then, in the dialogue stage, the DM is criticizing the proposed solutions unless one of them is completely satisfactory and Pareto efficient. In the latter case the procedure stops. Otherwise, the critic evaluation of proposed solutions is used as preference information to build a preference model of the DM. This model is used to calculate a new subset of feasible solutions in the next computation stage, with the intention to better fit the DM's preferences. In some procedures, the preference model appearing between the dialogue stage and the computation stage is implicit. However, it is useful when it can be explicitly shown to the DM for her approval. For this, the preference model should be easily understandable, and the treatment of preference information leading to the model should be intelligible for the DM. The decision rules stemming from DRSA fulfill both these requirements. The IMO-DRSA methodology is presented below.

We assume that the interactive procedure is exploring the set of feasible solutions of a multiobjective optimization problem. Denoting by X the considered set of solutions and by $f_i : X \to \mathbf{R}, i = 1, \ldots, m$, the objective functions to be maximized, the interactive procedure is composed of the following steps:

Step 1. Generate a representative sample of feasible solutions.

Step 2. Present the sample to the DM.

Step 3. If the DM is satisfied with one solution from the sample and the solution is Pareto optimal, then this is the compromise solution and the procedure stops. Otherwise continue.

Step 4. Ask the DM to indicate a subset of relatively "good" solutions in the sample.

Step 5. Apply DRSA to the current sample of solutions sorted into "good" and "others", in order to induce a set of decision rules with the following syntax: "*if* $f_{i_1}(\mathbf{x}) \geq \alpha_{i_1}$ *and* ... *and* $f_{i_p}(\mathbf{x}) \geq \alpha_{i_p}$, *then the solution is good*", $\{i_1, \ldots, i_p\} \subseteq \{1, \ldots, m\}$.

Step 6. Present the obtained set of rules to the DM.

Step 7. Ask the DM to select the most important decision rule for her in the set.

Step 8. Add the constraints $f_{i_1}(\mathbf{x}) \geq \alpha_{i_1}$, ..., $f_{i_p}(\mathbf{x}) \geq \alpha_{i_p}$ coming from the condition part of the rule selected in *Step* 7 to the set of constraints of the optimization problem at hand, in order to focus on a more interesting region of feasible solutions from the point of view of DM's preferences.

Step 9. Go back to *Step* 1.

The syntax of rules induced in *Step* 5 corresponds to maximization of objective functions. In case of minimization of objective function f_i, the corresponding condition would be $f_i(\mathbf{x}) \leq \alpha_i$.

Reduction of the set of feasible solutions cannot be considered as irreversible. Indeed, the DM can retract to the set of feasible solutions considered in one of previous iterations and continue from this point. This is in the spirit of a learning oriented conception of multiobjective interactive optimization, i.e. it agrees with the idea that the interactive procedure permits the DM to learn about her preferences and about the "shape" of the set of feasible solutions, and, in consequence, to correct previous decisions.

3 The Bound-and-Cut Algorithm

Let us consider the following problem $P1$:

$$\max f_1(x_1, \ldots, x_n)$$
$$\ldots$$
$$\max f_m(x_1, \ldots, x_n)$$

subject to constraints

$$g_1(x_1, \ldots, x_n) \leq 0$$
$$\ldots$$
$$g_p(x_1, \ldots, x_n) \leq 0$$
$$x_1, \ldots, x_r \in \mathbf{Z} \ (\mathbf{Z} \text{ is the set of integers}),$$
$$x_{r+1}, \ldots, x_n \in \mathbf{R}.$$

Our approach is based on the following simple observation. Let us suppose that there are $b_1, \ldots, b_m \in \mathbf{R}$, such that there exists a solution $\overline{\mathbf{x}} = (\overline{x}_1, \ldots, \overline{x}_n)$ satisfying, apart from the constraints of above problem $P1$, the following other constraints $C(b_1, \ldots, b_m)$:

$$f_1(x_1,\ldots,x_n) \geq b_1$$
$$\ldots$$
$$f_m(x_1,\ldots,x_n) \geq b_m.$$

Then, for each $(x_1,\ldots,x_n)$ in the Pareto set of problem $P1$ that satisfies constraints $C(b_1,\ldots,b_m)$, we have:

$$x_1 \in [x_1^l, x_1^u] \cap \mathbf{Z}, ..., x_r \in [x_r^l, x_r^u] \cap \mathbf{Z}$$

with $x_i^l = min\ x_i$ and $x_i^u = max\ x_i$, subject to the constraints of $P1$, with the exclusion of the integrality constraints (i.e. $x_1,\ldots,x_r \in \mathbf{R}$ instead of $x_1,\ldots,x_r \in \mathbf{Z}$), and to constraints $C(b_1,\ldots,b_m)$.

Based on the above observation, the closer the vector $(b_1,\ldots,b_m)$ to the Pareto front, the greater the reduction of the set of considered feasible solutions, and the greater the chance that combinations of possible values of integer variables $x_1,\ldots,x_r$ become manageable. The following example illustrates this point.

Example 1. Let us suppose that we are dealing with the following multiobjective mixed-integer optimization problem:

$$\begin{aligned}
max\ f_1(x_1,x_2,x_3,x_4,x_5) &= x_1 + 3x_2 - 2x_3 + 4x_4 - 3x_5\\
max\ f_2(x_1,x_2,x_3,x_4,x_5) &= 3x_1 - x_2 + 4x_3 - 2x_4 + 5x_5\\
max\ f_3(x_1,x_2,x_3,x_4,x_5) &= -4x_1 + 4x_2 + 3x_3 + 5x_4 - x_5\\
max\ f_4(x_1,x_2,x_3,x_4,x_5) &= 3x_1 + 5x_2 - x_3 + 7x_4 - 4x_5
\end{aligned}$$

subject to the following constraints

$$\begin{aligned}
g_1(x_1,x_2,x_3,x_4,x_5) &= 3x_1 + 5x_2 + 6x_3 - 2x_4 - x_5 - 15 \leq 0\\
g_2(x_1,x_2,x_3,x_4,x_5) &= -x_1 - x4_2 - 3x_3 + 4x_4 + 8x_5 - 90 \leq 0\\
g_3(x_1,x_2,x_3,x_4,x_5) &= 2x_1 + 6x_2 + 7x_3 + 8x_4 + 2x_5 - 100 \leq 0\\
g_4(x_1,x_2,x_3,x_4,x_5) &= x_1 - 4x_2 - 7x_3 + 2x_4 + 5x_5 - 30 \leq 0\\
g_5(x_1,x_2,x_3,x_4,x_5) &= -x_1 \leq 0\\
g_6(x_1,x_2,x_3,x_4,x_5) &= -x_2 \leq 0\\
g_7(x_1,x_2,x_3,x_4,x_5) &= -x_3 \leq 0\\
g_8(x_1,x_2,x_3,x_4,x_5) &= x_1 \leq 1\\
g_9(x_1,x_2,x_3,x_4,x_5) &= x_2 \leq 2\\
g_{10}(x_1,x_2,x_3,x_4,x_5) &= x_3 \leq 3
\end{aligned}$$

with $x_1, x_2, x_3 \in \mathbf{Z}$ and $x_4, x_5 \in \mathbf{R}$. Observe that for $\overline{\mathbf{x}} = [1,2,2,5,7]$ we have

$$f_1(\overline{\mathbf{x}}) = 2, f_2(\overline{\mathbf{x}}) = 34, f_3(\overline{\mathbf{x}}) = 28, f_4(\overline{\mathbf{x}}) = 22$$

and all the constraints of the above optimization problem are satisfied. Thus, let us compute the maximum and the minimum values for the integer variables x_1, x_2, x_3, under the constraints of the original optimization problem plus the constraints:

$$f_1(x_1,x_2,x_3,x_4,x_5) = x_1 + 3x_2 - 2x_3 + 4x_4 - 3x_5 \geq 2,$$

$$\begin{gathered}f_2(x_1,x_2,x_3,x_4,x_5) = 3x_1 - x_2 + 4x_3 - 2x_4 + 5x_5 \geq 34,\\ f_3(x_1,x_2,x_3,x_4,x_5) = -4x_1 + 4x_2 + 3x_3 + 5x_4 - x_5 \geq 28,\\ f_4(x_1,x_2,x_3,x_4,x_5) = 3x_1 + 5x_2 - x_3 + 7x_4 - 4x_5 \geq 22,\end{gathered}$$

while relaxing the integrality constraints. We get the following results:

$$x_1^u = max\ x_1 = 1, x_2^u = max\ x_2 = 2, x_3^u = max\ x_3 = 3,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 0, x_3^l = min\ x_3 = 1.58,$$

such that variables x_1 and x_2 maintain the whole initial range of variation (i.e. x_1 can assume values 0 and 1, and x_2 can assume values 0, 1 and 2), while the variable x_3 reduces the initial range of variation and can assume values 2 and 3 (instead of the initial range of possible values being 0,1,2 and 3), which amounts to 12 possible combinations of values for the three integer variables x_1, x_2 and x_3. Observe that not all of these 12 combinations of values are admissible in the sense that for some of them there are no values for x_4 and x_5 such that all the constraints $C(2, 34, 28, 22)$ are satisfied. For example, the combination of values $x_1 = 0, x_2 = 0, x_3 = 2$ is not admissible.

Let us try to reduce further the range of variation of the integer variables by considering some tighter bounds on the objective functions f_1, f_2, f_3, f_4. Observe that for $\overline{\mathbf{x}} = [0, 2, 3, 6.5, 7.5]$ we have

$$f_1(\overline{\mathbf{x}}) = 3.5, f_2(\overline{\mathbf{x}}) = 34.5, f_3(\overline{\mathbf{x}}) = 42, f_4(\overline{\mathbf{x}}) = 28.5,$$

and all the constraints of the above optimization problem are satisfied. Thus, one could decide to investigate a part of the set of feasible solutions of above problem $P1$, such that

$$f_1(\mathbf{x}) \geq 3, f_2(\mathbf{x}) \geq 34, f_3(\mathbf{x}) \geq 40, f_4(\mathbf{x}) \geq 25.$$

Notice that the bounds considered for the objective functions are weaker than the values taken by f_1, f_2, f_3, f_4 for $\overline{\mathbf{x}} = [0, 2, 3, 6.5, 7.5]$ in order to explore a larger part of the set of feasible solutions and not to reduce it too early. Thus, for investigating the range of variation of the integer variables x_1, x_2, x_3 in the considered part of the set of feasible solutions, let us compute the maximum and the minimum values under the constraints of the original optimization problem, plus the constraints:

$$\begin{gathered}f_1(x_1,x_2,x_3,x_4,x_5) = x_1 + 3x_2 - 2x_3 + 4x_4 - 3x_5 \geq 3,\\ f_2(x_1,x_2,x_3,x_4,x_5) = 3x_1 - x_2 + 4x_3 - 2x_4 + 5x_5 \geq 34,\\ f_3(x_1,x_2,x_3,x_4,x_5) = -4x_1 + 4x_2 + 3x_3 + 5x_4 - x_5 \geq 40,\\ f_4(x_1,x_2,x_3,x_4,x_5) = 3x_1 + 5x_2 - x_3 + 7x_4 - 4x_5 \geq 25,\end{gathered}$$

while relaxing the integrality constraints. We get the following results:

$$x_1^u = max\ x_1 = 0.58, x_2^u = max\ x_2 = 2, x_3^u = max\ x_3 = 3,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 0.30, x_3^l = min\ x_3 = 2.24,$$

such that the three integer variables further reduce their range of variation. x_1 can assume only value 0, x_2 can assume values 1 or 2, and x_3 can assume only value 3. Thus, only 2 possible combinations of values for the three integer variables x_1, x_2 and x_3 remain: $x_1 = 0$, $x_2 = 1$ and $x_3 = 3$, and $x_1 = 0$, $x_2 = 2$ and $x_3 = 3$. Both these combinations of values of integer variables are admissible. **End of example 1.**

Coming back to our first observation, we can weaken it by considering a subset of objective functions. More precisely, instead of considering constraints $C(b_1, \ldots, b_m)$ concerning lower bounds b_j on all the objective functions f_j, $j = 1, \ldots, m$, we can consider constraints $C_J(b_{j_1}, \ldots, b_{j_p})$ for objective functions $f_{j_1}, \ldots, f_{j_p}$, with $J = \{j_1, \ldots, j_p\} \subseteq \{1, \ldots, m\}$. The following example illustrates this point.

Example 2. In example 1, we have already observed that for $\overline{\mathbf{x}} = [0, 2, 3, 6.5, 7.5]$ we have

$$f_1(\overline{\mathbf{x}}) = 3.5, f_2(\overline{\mathbf{x}}) = 34.5, f_3(\overline{\mathbf{x}}) = 42, f_4(\overline{\mathbf{x}}) = 28.5,$$

and all the constraints of the above optimization problem are satisfied. Now, let us try to investigate the portion of the set of feasible solutions of above problem $P1$, such that

$$f_2(\mathbf{x}) \geq 34, f_3(\mathbf{x}) \geq 40.$$

In this case, we are considering a subset of objective functions, more precisely f_2 and f_3. To investigate the range of variation of the integer variables x_1, x_2, x_3 in the considered part of the set of feasible solutions, let us compute the maximum and the minimum values under the constraints of the original optimization problem, plus the constraints:

$$f_2(x_1, x_2, x_3, x_4, x_5) = 3x_1 - x_2 + 4x_3 - 2x_4 + 5x_5 \geq 34,$$
$$f_3(x_1, x_2, x_3, x_4, x_5) = -4x_1 + 4x_2 + 3x_3 + 5x_4 - x_5 \geq 40,$$

while relaxing the integrality constraints. We get the following results:

$$x_1^u = max\ x_1 = 0.58, x_2^u = max\ x_2 = 2, x_3^u = max\ x_3 = 3,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 0.28, x_3^l = min\ x_3 = 2.24,$$

such that the three integer variables further reduce their range of variation. x_1 can assume only value 0, x_2 can assume value 1 or 2, and x_3 can assume only value 3. Thus, considering objective functions f_2 and f_3, only 2 possible combinations of values for the three integer variables x_1, x_2 and x_3 remain: $x_1 = 0$, $x_2 = 1$ and $x_3 = 3$, and $x_1 = 0$, $x_2 = 2$ and $x_3 = 3$. **End of example 2.**

Let us observe that the idea considered in examples 1 and 2 can also be applied to classical single objective optimization, as shown in the following example.

Example 3. Let us suppose that we want to maximize f_1 only, subject to the constraints of the above problem $P1$. Observe that for $\overline{\mathbf{x}} = [0, 0, 1, 27.75, -64.5]$ we have $f_1(\overline{\mathbf{x}}) = 302.5$. For each integer variable let us compute the maximum and the minimum values under the constraints of the original optimization problem, plus the constraint:

$$f_1(x_1, x_2, x_3, x_4, x_5) = x_1 + 3x_2 - 2x_3 + 4x_4 - 3x_5 \geq 302.5,$$

while relaxing the integrality constraints. We get the following results:

$$x_1^u = max\ x_1 = 1, x_2^u = max\ x_2 = 1.30, x_3^u = max\ x_3 = 1,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 0, x_3^l = min\ x_3 = 0.$$

Thus, we can conclude that the three integer variables reduce their range of variation and, more precisely, all the three variables can assume values 0 or 1. Not all possible combinations of values of the three integer variables are admissible (for example, $x_1 = 1, x_2 = 1$ and $x_3 = 0$ is a non admissible combination). By investigation of the remaining part of the variable space, one can observe that for $\overline{\mathbf{x}} = [1, 0, 0, 30.50, -73]$ we have $f_1(\overline{\mathbf{x}}) = 342$. If for each integer variable, we compute again the maximum and the minimum values under the constraints of the original optimization problem, plus the constraint:

$$f_1(x_1, x_2, x_3, x_4, x_5) = x_1 + 3x_2 - 2x_3 + 4x_4 - 3x_5 \geq 342,$$

while relaxing the integrality constraints, we get the following results:

$$x_1^u = max\ x_1 = 1, x_2^u = max\ x_2 = 0.54, x_3^u = max\ x_3 = 0.41,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 0, x_3^l = min\ x_3 = 0.$$

Thus, we can conclude that the integer variables further reduced their range of variation, such that x_1 can assume value 0 or 1, and x_2 and x_3 can assume only value 0. Consequently, only two combinations of values for integer variables are possible: $x_1 = 0$, $x_2 = 0$ and $x_3 = 0$, and $x_1 = 1$, $x_2 = 0$ and $x_3 = 0$. Checking these two combinations, one can see that both of them are admissible, that the optimal solution is $\mathbf{x} = [0, 0, 0, 32.5, -80]$, and the optimal value of the objective function is $f_1(0, 0, 0, 32.5, -80) = 370$. **End of example 3.**

Taking into account examples 1, 2 and 3, we can propose a general algorithm to deal with mixed-integer optimization, that we call "*bound-and-cut*". The name refers to the fact that the algorithm alternates steps in which a lower bound for the objective function(s) is fixed, with steps in which the range of variation of the integer variables is cut, taking into account their upper and lower bounds, and excluding the integrality constraints. The lower bounds on the objective function(s) permit to reduce the range of variation of the integer variables, and the upper and lower bounds on integer variables permit to concentrate the search for Pareto-optimal solutions in more and more promising regions.

Efficient application of the bound-and-cut algorithm is based on the answers to the following questions:

1. How to determine one or more solutions $(\overline{x}_1, \ldots, \overline{x}_n)$ satisfying constraints of problem $P1$, and located close enough to the Pareto front?
2. How to use the reduction of the range of variation of the integer variables?
3. Having determined one or more solutions $(\overline{x}_1, \ldots, \overline{x}_n)$, how to select the objective functions and how to fix the bounds b_j to define the set of constraints $C_J(b_{j_1}, \ldots, b_{j_p})$?

The answer to the first question is quite easy. One can fix the value of $m-1$ objective functions, and optimize the remaining one, using any single objective mixed-integer optimization method, like "branch-and-bound", "branch-and-cut" or "branch-and-price". In this way, one can find a Pareto optimal solution of the considered problem that can be taken as a starting point for the bound-and-cut procedure. One can observe that methods of discrete optimization are very often computationally expensive and perhaps their application within the bound-and-cut algorithm would make the whole approach inefficient. Let us remark, however, that one does not need necessarily a solution which is Pareto optimal, because it is sufficient that the solution satisfies constraints of problem $P1$ and is located "not too far" from the Pareto front. In these circumstances, it seems appropriate to search for such a solution using an evolutionary heuristic.

The answer to the second question is also easy. In fact, the reduction of the range of variation of integer variables simplifies a lot the application of any exact or heuristic method used to solve problem $P1$. Let us observe that the possibility of reducing the range of variation of integer variables is also useful for single objective optimization problems known for combinatorial explosion of the solution space.

In the context of arguments given in section 2, the answer to the third question is quite natural: one can use the Interactive Multiobjective Optimization driven by Dominance-based Rough Approach (IMO-DRSA). In the dialogue stage, IMO-DRSA gives a representation of DM's preferences in terms of decision rules, whose syntax is: "if $f_{j_1}(\mathbf{x}) \geq b_{j_1}$ and ... and $f_{j_p}(\mathbf{x}) \geq b_{j_p}$, then solution $\mathbf{x}$ is good". In fact, each one of these decision rules defines a subset of objective functions and a corresponding set of constraints $C_J(b_{j_1}, \ldots, b_{j_p})$ that can be efficiently used within the bound-and-cut algorithm.

Some further remarks related to the use of the method in nonlinear programming problems, and about the coupling with an evolutionary optimization method can be useful. In fact, bound-and-cut needs to solve optimization problems in two steps of the methodology:

a) when some Pareto-optimal solutions have to be found in order to be presented to the DM, and
b) when, once fixed the lower bounds on the objective function to be maximized, the maximum and the minimum of each integer variable have to be determined.

For point a), in case of nonlinear integer (multiobjective) optimization problems, perhaps the best is to use some EMO heuristic to find a reasonable approximation of the Pareto set. Point b), instead, requires the solution of a single objective (non-integer) optimization problem (the single objective to be maximized or minimized is the integer variable, however, in this case it does not need to be integer): in this step, some classical method for single objective optimization, such as gradient descent or conjugate gradient, or even some evolutionary optimization method, can be efficiently used.

4 A Didactic Example

In this section, we shall present a didactic example illustrating the whole methodology. Let us consider the same problem as in example 1 of section 3. In the first iteration, after the calculation stage, we propose to the DM the sample of solutions presented in Table 1. The DM evaluates these solutions and indicates those which are relatively good in her opinion. This information is shown in the last column of Table 1.

Table 1. First sample of admissible solutions

Solutions	x_1	x_2	x_3	x_4	x_5	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_4(\mathbf{x})$	Overall evaluation
s_1	0	0	0	32.5	-80	370.00	-465.00	242.50	547.50	*
s_2	0	0	0	-13.13	11.25	-86.25	82.50	-76.88	-136.88	*
s_3	1	2	3	7.52	2.41	23.86	10.00	48.20	59.02	good
s_4	1	2	3	7.98	0.59	31.14	0.00	52.30	69.48	good
s_5	0	2	3	8.068	1.23	28.59	0.00	56.11	64.57	good
s_6	1	2	0	9.91	3.36	36.55	-2.00	50.18	68.91	*
s_7	1	2	3	6.26	7.46	3.67	37.76	36.85	30.00	good
s_8	1	2	0	12.09	-5.36	71.45	-50.00	69.82	119.09	good
s_9	1	0	3	13.47	-17.47	101.31	-99.31	89.83	170.19	*

The application of DRSA to information from Table 1 gives the following minimal set of decision rules covering all solutions with Overall Evaluation "good" (between parentheses the solutions from Table 1 supporting the decision rule):

rule 1.1) If $f_1(\mathbf{x}) \geq 23.86$ and $f_2(\mathbf{x}) \geq 10$, then the solution is good $\{s3\}$,
rule 1.2) If $f_1(\mathbf{x}) \geq 28.59$ and $f_2(\mathbf{x}) \geq 0$, then the solution is good $\{s4, s5\}$,
rule 1.3) If $f_2(\mathbf{x}) \geq 37.76$ and $f_3(\mathbf{x}) \geq 36.85$, then the solution is good $\{s7\}$,
rule 1.3) If $f_2(\mathbf{x}) \geq -50$ and $f_4(\mathbf{x}) \geq 119.09$, then the solution is good $\{s8\}$.

The DM selects rule 1.3 as the most representative of her preferences.

According to this selection, the following constraints are added to the original optimization problem:

$$f_2(x_1, x_2, x_3, x_4, x_5) = 3x_1 - x_2 + 4x_3 - 2x_4 + 5x_5 \geq 37.76,$$
$$f_3(x_1, x_2, x_3, x_4, x_5) = -4x_1 + 4x_2 + 3x_3 + 5x_4 - x_5 \geq 36.85.$$

Applying the bound-and-cut algorithm for investigating the range of variation of the integer variables x_1, x_2, x_3 in the considered part of the set of feasible solutions, let us compute their maximum and minimum values under the constraints of the original optimization problem, plus the new constraints, while relaxing the integrality constraints. We get the following results:

$$x_1^u = max\ x_1 = 1, x_2^u = max\ x_2 = 2, x_3^u = max\ x_3 = 3,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 0.55, x_3^l = min\ x_3 = 2.36.$$

Thus the three integer variables reduce their range of variation and, more precisely, x_1 can continue to assume value 0 or 1, x_2 can assume value 1 or 2, and x_3 can assume only value 3, such that only four possible combinations of values for the three integer variables x_1, x_2 and x_3 remain: 1) $x_1 = 0$, $x_2 = 1$, $x_3 = 3$; 2) $x_1 = 0$, $x_2 = 2$, $x_3 = 3$; 3) $x_1 = 1$, $x_2 = 1$, $x_3 = 3$; 4) $x_1 = 1$, $x_2 = 2$, $x_3 = 3$. Combination of values $x_1 = 1$, $x_2 = 1$, $x_3 = 3$ is not admissible and thus a new sample of representative solutions was generated taking into account the remaining three combinations of values. This sample is presented in the following Table 2. The DM evaluates the alternative solutions from Table 2 and indicates those which are relatively good, as shown in the last column of Table 2.

Table 2. Second sample of admissible solutions

Solutions	x_1	x_2	x_3	x_4	x_5	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_4(\mathbf{x})$	Overall evaluation
s_{10}	0	1	3	7.06	8.18	0.71	37.76	40.12	24.72	good
s_{11}	0	1	3	6.45	8.42	-2.44	40.19	36.85	19.50	*
s_{12}	0	2	3	6.35	8.09	1.13	37.76	40.67	25.09	good
s_{13}	0	2	3	5.86	9.46	-4.92	45.56	36.85	16.21	*
s_{14}	1	2	3	6.26	7.46	3.67	37.76	36.85	30.00	*
s_{15}	0	1	3	6.92	8.23	0.00	38.31	39.38	23.54	*
s_{16}	0	2	3	6.28	8.38	0.00	39.31	40.03	23.47	*
s_{17}	0	2	3	6.34	8.13	1.00	37.94	40.59	24.91	good

Using DRSA to preference information from Table 2, one obtains the following minimal decision rule covering all solutions from the second sample with Overall Evaluation "good":

rule 2.1) If $f_3(\mathbf{x}) \geq 40.12$, then the solution is good, $\{s_{10}, s_{12}, s_{17}\}$.

The DM considers rule 2.1 to be representative of her preferences and, consequently, the following constraint is added to the original optimization problem:

$$f_3(x_1, x_2, x_3, x_4, x_5) = -4x_1 + 4x_2 + 3x_3 + 5x_4 - x_5 \geq 40.12.$$

Applying again the bound-and-cut algorithm for investigating the range of variation of the integer variables x_1, x_2, x_3 in the considered part of the set of feasible solutions, let us compute their maximum and minimum values under the previous constraints of the optimization problem, plus the new constraints, while relaxing the integrality constraints. We get the following results:

$$x_1^u = max\ x_1 = 0.14, x_2^u = max\ x_2 = 2, x_3^u = max\ x_3 = 3,$$

$$x_1^l = min\ x_1 = 0, x_2^l = min\ x_2 = 1, x_3^l = min\ x_3 = 2.56.$$

Thus, the three integer variables reduce their range of variation and more precisely, x_1 can assume only value 0, x_2 can assume value 1 or 2, and x_3 can assume only value 3, such that only 2 possible combinations of values for the three integer variables x_1, x_2 and x_3 remain: 1) $x_1 = 0$, $x_2 = 1$, $x_3 = 3$; 2) $x_1 = 0$, $x_2 = 2$, $x_3 = 3$. Both these two combinations of values of integer variables are admissible and, consequently, a new sample of representative solutions was generated taking into account those two combinations of values. This sample is presented in the following Table 3. The DM evaluates the alternative solutions from Table 3 and selects solution s_{21}.

Table 3. Third sample of admissible solutions

Solutions	x_1	x_2	x_3	x_4	x_5	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_4(\mathbf{x})$	Overall evaluation
s_{18}	0	1	3	7.06	8.18	0.71	37.76	40.12	24.71	*
s_{19}	0	2	3	6.35	8.09	1.13	37.76	40.67	25.09	*
s_{20}	0	2	3	6.29	8.34	0.16	39.10	40.12	23.70	*
s_{21}	0	2	3	6.34	8.13	1.00	37.94	40.59	24.91	selected
s_{22}	0	2	3	6.32	8.23	0.59	38.50	40.36	24.32	*
s_{23}	0	2	3	6.30	8.32	0.23	39.00	40.16	23.80	*
s_{24}	0	2	3	6.30	8.28	0.37	38.80	40.24	24.00	*
s_{25}	0	2	3	6.33	8.19	0.74	38.30	40.45	24.53	*

5 Conclusions

We presented a new methodology for dealing with interactive multiobjective optimization in case of mixed-integer optimization. Our approach is based on a new algorithm for handling integer variables in optimization problems. This new algorithm is called bound-and-cut. It is using some lower (upper) bounds on the objective functions to be maximized (minimized), which are fixed taking into account preferences of the DM. On one hand, these bounds gradually constrain the set of feasible solutions ensuring convergence to the most preferred solution on the Pareto front, and on the other hand, they reduce step by step the range of variation of the integer variables until the most interesting part of the Pareto front is small enough to be exhaustively explored by the DM with respect to feasible values of the integer variables. The bound-and-cut algorithm is very general and can be used also in case of classical mixed-integer optimization problems involving only one objective function. The bound-and-cut algorithm can be efficiently conjugated with an Evolutionary Multiobjetive Optimization (EMO) method. In the context of the interactive multiobjective optimization, we proposed to integrate the bound-and-cut algorithm within the IMO-DRSA methodology because, besides its characteristics of transparency and good understandability, it gives a preference model expressed in terms of decision rules providing explicitly the lower (upper) bounds on the objective functions to be maximized (minimized) required by the bound-and-cut algorithm.

Acknowledgment. The third author wishes to acknowledge financial support from the Polish Ministry of Science and Higher Education, grant N N519 441939.

References

1. Alves, M.J., Climaco, J.: A review of interactive methods for multiobjective integer and mixed-integer programming. European Journal of Operatinal Research 180, 99–115 (2007)
2. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-Price: Column Generation for Huge Integer Programs. Operations Research 46(3), 316–329 (1998)
3. Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): Multiobjective Optimization. LNCS, vol. 5252. Springer, Heidelberg (2008)
4. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)
5. Evans, G.W.: An overview of techniques for solving multi-objective mathematical programs. Management Science 30, 1263–1282 (1984)
6. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Mathematical Society 64, 275–278 (1958)
7. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129, 1–47 (2001)
8. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-Based Rough Set Approach to Interactive Multiobjective Optimization. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization: Interactive and Evolutionary Approaches. LNCS, vol. 5252, pp. 121–155. Springer, Heidelberg (2008)
9. Land, A.H., Doig, A.G.: An automatic method for solving discrete programming problems. Econometrica 28, 497–520 (1960)
10. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
11. Słowiński, R., Greco, S., Matarazzo, B.: Rough set based decision support. In: Burke, E.K., Kendall, G. (eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, ch. 16, pp. 475–527. Springer, Heidelberg (2005)
12. Słowiński, R., Greco, S., Matarazzo, B.: Rough Sets in Decision Making. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7753–7786. Springer, Heidelberg (2009)

Very Large-Scale Neighborhood Search for Solving Multiobjective Combinatorial Optimization Problems

Thibaut Lust, Jacques Teghem, and Daniel Tuyttens

Faculté Polytechnique de Mons, Laboratory of Mathematics & Operational Research
20, place du parc 7000 Mons, Belgium
lust.thibaut@gmail.com

Abstract. Very large-scale neighborhood search (VLSNS) is a technique intensively used in single-objective optimization. However, there is almost no study of VLSNS for multiobjective optimization. We show in this paper that this technique is very efficient for the resolution of multiobjective combinatorial optimization problems. Two problems are considered: the multiobjective multidimensional knapsack problem and the multiobjective set covering problem. VLSNS are proposed for these two problems and are integrated into the two-phase Pareto local search. The results obtained on biobjective instances outperform the state-of-the-art results for various indicators.

1 Introduction

During the last 20 years, many heuristic methods for solving multiobjective combinatorial optimization (MOCO) problems have been proposed. From the first survey [33] in 1994 till [13] in 2002, a lot of papers have been published and this flow is still increasing. The main reason of this phenomenon is the success story of metaheuristics [15].

Effectively, it is quite difficult to determine exactly the whole set of Pareto efficient solutions X_E and the set of Pareto non-dominated points Y_N for MOCO problems. This is a $\mathcal{NP}$-Hard problem even for MOCO problems for which polynomial algorithms exist for the single-objective counterpart, such as the linear assignment problem. Therefore, there exist only few exact methods able to determine the sets X_E and Y_N and we can only expect to apply these methods for small instances and for few number of objectives. For this reason, many methods are heuristic methods which produce approximations $\widetilde{X}_E$ and $\widetilde{Y}_N$ to the sets X_E and Y_N. Due to the success of metaheuristics for single-objective CO, multiobjective metaheuristics became quickly a classic tool to tackle MOCO problems and it is presently a real challenge for the researchers to improve the results previously obtained.

If evolutionary methods [9,12] and simple local search [13] (LS) have intensively been applied to solve multiobjective problems (MOPs), few results are known about the use of elaborate LS techniques, as very large-scale neighborhood search

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 254–268, 2011.

(VLSNS) [1] or variable neighborhood search (VNS) [16] for solving MOPs. It is mainly because it is more natural to use a method based on a population, as we are looking for an approximation to a set. However, by embedding these evolved LS techniques into the two-phase Pareto local search (2PPLS) [28], which uses as population the set of potentially efficient solutions, we show in this paper that we can produce very effective heuristics in order to solve MOCO problems.

The paper is organized as follows. In the next section, we present VLSNS for the multiobjective multidimensional knapsack problem (MOMKP) and for the multiobjective set covering problem (MOSCP). We show after how VLSNS can be embedded into 2PPLS. We finally present the results obtained for the two MOCO problems considered, with a comparison to state-of-the-art results.

2 Very Large-Scale Neighborhood Search

With LS, the larger the neighborhood, the better the quality of the local optimum obtained is. However, by increasing the size of the neighborhood, the time to explore the neighborhood becomes higher. Therefore, using a larger neighborhood does not necessary give rise to a more effective method. If we want to keep reasonable running times while using a large neighborhood, an efficient strategy has to be implemented in order to explore the neighborhood; this is what is done in VLSNS.

VLSNS is very popular in single-objective optimization [1]. For example, the Lin-Kernighan heuristic [25], one of the best heuristics for solving the single-objective traveling salesman problem (TSP), is based on VLSNS. On the other hand, there is almost no study of VLSNS for solving MOCO problems. The only known result is the LS of Angel *et al.* [6], which integrates a dynasearch neighborhood (the neighborhood is solved with dynamic programming) to solve the biobjective TSP.

We present VLSNS for two problems: the MOMKP and the MOSCP. Starting from a current solution, called x^c, the aim of VLSNS is to produce a set of neighbors of high quality, in a reasonable time. The technique that we use for solving the MOMKP or the MOSCP is the following:

1. Identification of a set of variables candidates to be removed from x^c (set $\mathcal{O}$)
2. Identification of a set of variables, not in x^c, candidates to be added (set $\mathcal{I}$)
3. Creation of a residual multiobjective problem formed by the variables belonging to $\{\mathcal{O} \cup \mathcal{I}\}$.
4. Resolution of the residual problem: a set of potentially efficient solutions of this problem is produced. The potentially efficient solutions are then merged with the unmodified variables of x^c to produce the neighbors.

We present now how we have adapted this technique to the two MOCO problems considered in this paper.

2.1 The Multiobjective Multidimensionnal Knapsack Problem

The MOMKP consists in determining a subset of items, among n items ($i = 1, \ldots, n$) having m characteristics w^i_j ($j = 1, \ldots, m$) and p profits c^i_l

$(l = 1, \ldots, p)$, such that the p total profits are maximized and the m knapsack capacities W_j regarding the different characteristics are respected.

The formulation is the following:

$$\begin{aligned} \text{“max”}\ f_l(x) &= \sum_{i=1}^{n} c_l^i x_i & l &= 1, \ldots, p \\ \text{subject to } \sum_{i=1}^{n} w_j^i x_i &\leq W_j & j &= 1, \ldots, m \\ x_i &\in \{0, 1\} & i &= 1, \ldots, n \end{aligned}$$

where $x_i = 1$ means that the item i is selected to be in the knapsack. It is assumed that all coefficients c_l^i, w_j^i and W_j are nonnegative.

According to the general technique previously presented to define VLSNS, the different steps to produce neighbors from a current solution x^c of the MOMKP are as follows:

1. The set $\mathcal{O}$ of items candidates to be removed will be composed of the k worst items (present in x) for the ratio R_1, defined by

$$R_1 = \frac{\sum_{l=1}^{p} \lambda_l c_l^s}{\sum_{j=1}^{m} w_j^s} \tag{1}$$

   for an item s. This ratio is simply equal to the weighted linear aggregation of the profits on the sum of the weights of the item s.

2. The set $\mathcal{I}$ of items candidates to be added will be composed of the k best items (not in x) for the ratio R_2 [30], defined by

$$R_2 = \frac{\sum_{l=1}^{p} \lambda_l c_l^s}{\sum_{j=1}^{m} \left(\frac{w_j^s}{W_j - \sum_{\substack{i=1 \\ i \notin \mathcal{O}}}^{n} w_j^i x_i + 1} \right)} \tag{2}$$

   for an item s.

   This ratio is also used in MOGLS [21], and consists in selecting items of high profit and low weight, by giving higher influence to the constraints whose the values (measured by $\sum_{\substack{i=1 \\ i \notin \mathcal{O}}}^{n} w_j^i x_i$) is close to the maximum capacity W_j.

3. A residual problem is defined, of size $(k * 2)$, and composed of the items belonging to the set $\{\mathcal{O} \cup \mathcal{I}\}$. The capacities W_j^r of the residual problem are equal to $W_j - \sum_{\substack{i=1 \\ i \notin \mathcal{O}}}^{n} w_j^i x_i$ with $j = 1, \ldots, m$.

4. To solve the residual problem, we have employed the multiobjective metaheuristic MEMOTS [26], a memetic algorithm integrating tabu search, that has already been applied to the MOMKP. MEMOTS presents some parameters that we will not tune here; we will rather take into account the conclusions obtained in [26]. We will only study the number of iterations N performed in MEMOTS to solve the residual problems.

   Once the residual problem has been solved, we merge the potentially efficient solutions of the residual problem with the unmodified variables of x^c, to obtain the neighbors.

2.2 The Multiobjective Set Covering Problem

In the MOSCP, we have a set of m items, and each item can be covered by a subset of n sets. Each set j has p costs c_l^j $(l = 1, \ldots, p)$. The MOSCP consists in determining a subset of sets, among the n sets $(j = 1, \ldots, n)$ such that all the items are covered by at least one set and that the p total costs are minimized.

The formulation is the following:

$$\begin{array}{lll} \text{“min”} & f_l(x) = \sum_{j=1}^{n} c_l^j x_j & l = 1, \ldots, p \\ \text{subject to} & \sum_{j=1}^{n} t_{ij} x_j \geq 1 & i = 1, \ldots, m \\ & x_j \in \{0, 1\} & j = 1, \ldots, n \end{array}$$

with x the decision vector, formed of the binary variables x_j ($x_j = 1$ means that the set j is considered), and the binary data t_{ij} equal to 1 if the set j covers the item i and 0 otherwise.

It is assumed that all coefficients c_l^j are nonnegative.

The VLSNS that we have defined for the MOSCP is similar to the one defined for the MOMKP. However, the definition is a bit more complicated. With the MOMKP, the two sets $\mathcal{O}$ and $\mathcal{I}$ were independently defined. With the MOSCP, that cannot be anymore the case. Indeed, when we remove some sets of the current solution x^c, some items are not anymore covered. The set $\mathcal{I}$ must thus be composed of sets than can cover these items. Also, we have noticed that removing a small number k of sets (less than 4, as we will see later in Figure 3) from x^c was enough. Indeed, when we work with a larger set $\mathcal{O}$, it becomes more difficult to define the set $\mathcal{I}$ since many items become uncovered.

Therefore, as we will keep k small, for each value of k, we will create more than one residual problem. If $k = 1$, the number of residual problems will be equal to the number of sets present in the current solution x. For each residual problem, the set $\mathcal{O}$ will be thus composed of one of the sets present in x^c.

If $k > 1$, we create a set $\mathcal{L}$ of size L $(L \geq k)$ that includes the interesting sets to remove from x^c. Then, all the combinations of k sets from $\mathcal{L}$ will be considered to form the set $\mathcal{O}$. The number of residual problems will be thus equal to the number of combinations of k elements into a set of size L, that is C_k^L. The size of the set $\mathcal{I}$ will also be limited to L.

More precisely, the VLSNS works as follows:

1. If $k > 1$, the set $\mathcal{O}$ will be composed of k sets chosen among the list $\mathcal{L}$ containing the L worst sets (present in x^c) for the ratio R_3, defined by

$$R_3 = \frac{\sum_{l=1}^{p} \lambda_l c_l^s}{\sum_{i=1}^{m} t_{is}} \tag{3}$$

   for a set s. This ratio is equal to the weighted aggregation of the costs on the number of items that the set s covers.
2. The set $\mathcal{I}$ of sets candidates to be added will be composed of the L best sets (not in x) for the ratio R_4, defined by

$$R_4 = \frac{\sum_{l=1}^{p} \lambda_l c_l^s}{n_s} \tag{4}$$

   for a set s, where n_s represent the number of items that the set s covers among the items that are not anymore covered following the removals of the k preceding sets. This ratio is only computed for the sets that cover at least one of the items not anymore covered ($n_s > 0$).
3. A residual problem is defined, of size $k + L$, and composed of the sets belonging to the set $\{\mathcal{O} \cup \mathcal{I}\}$.
4. To solve the residual problem, we simply generate all the efficient solutions of the problem, with an enumeration algorithm. We then merge the efficient solutions of the residual problem with the unmodified variables of x^c, to obtain the neighbors.

3 Two-Phase Pareto Local Search

In order to be able to use VLSNS in MO, we have integrated this technique into the two-phase Pareto local search (2PPLS) [28]. As indicated by its name, the method is composed of two phases. In the first phase, an initial population of potentially efficient solutions is produced. In the second phase, the Pareto local search [6,31] (PLS) is run from this population. PLS is a straightforward adaptation of LS to MO and only needs a neighborhood function $\mathcal{N}(x)$, which is applied to every new potentially non-dominated solution generated. At the end, a local optimum, defined in a MO context, is obtained [31] (called a Pareto local optimum set). Therefore, to adapt 2PPLS to a MOCO problem, we have to define an initial population and a neighborhood function. As neighborhood, we will use a VLSNS.

However, comparing to the initial version of 2PPLS [28], we have adapted the method in order to be able to use different sizes of neighborhood, through a variable neighborhood search (VNS) technique [16]. This is particularly useful in the case of the MOSCP. Indeed, for the MOSCP, the VLSNS of size $(k+1)$ will not necessary produce the neighbors obtained with the VLSNS of size k (as the set $\mathcal{I}$ depends on the set $\mathcal{O}$ of size k). On the other hand, for the MOMKP, the VLSNS of size $(k+1)$ will also produce the neighbors generated by the VLSNS of size k. Therefore, for the MOMKP, the higher k, the better the results are, and a VNS technique is not worthwhile.

The pseudo-code of 2PPLS with VNS is given by the Algorithm 1.

The method needs four parameters: an initial population P_0, the size of the smallest ($k_{\min}$) (respectively largest ($k_{\max}$)) neighborhood structure, and the different neighborhood functions $\mathcal{N}_k(x)$ for each $k \in \mathbb{Z} : k_{\min} \leq k \leq k_{\max}$.

The method starts with the population P composed of potentially efficient solutions given by the initial population P_0. The neighborhood structure initially used is the smallest ($k = k_{\min}$). Then, considering $\mathcal{N}_k(x)$, all the neighbors p' of each solution p of P are generated. If a neighbor p' is not weakly Pareto dominated ($\not\preceq_P$) by the current solution p, we try to add the solution p' to the approximation $\widetilde{X}_E$ of the efficient set, which is updated with the procedure `AddSolution`. This procedure is not described in this paper but simply consists of updating an approximation $\widetilde{X}_E$ of the efficient set when a new solution p' is added to $\widetilde{X}_E$. This procedure has four parameters: the set $\widetilde{X}_E$ to actualize, the new solution p', its evaluation $f(p')$ and a boolean variable called *Added* that returns *True* if the new solution has been added and *False* otherwise. If the solution p' has been added to $\widetilde{X}_E$, *Added* is true and the solution p' is added to an auxiliary population P_a, which is also updated with the procedure `AddSolution`. Therefore, P_a is only composed of new potentially efficient solutions. Once all the neighbors of each solution of P have been generated, the algorithm starts again, with P equal to P_a, until $P = P_a = \varnothing$. The auxiliary population P_a is used such that the neighborhood of each solution of P is explored, even if some solutions of P become dominated following the addition of a new solution to P_a. Thus, sometimes, neighbors are generated from a dominated solution.

In the case of $P = P_a = \varnothing$, the set $\widetilde{X}_E$ obtained is a Pareto local optimum set according to $\mathcal{N}_k$, and cannot be thus improved with $\mathcal{N}_k$. We then increase the size of the neighborhood ($k \leftarrow k+1$), and apply again PLS with this larger neighborhood.

Please note that, in general, a solution Pareto local optimum for the neighborhood k is not necessary Pareto local optimum for the neighborhood $(k-1)$. That is why, after considering a larger neighborhood, we always restart the search with the smallest neighborhood structure.

After the generation of a Pareto local optimum set according to $\mathcal{N}_k$, the population P used with the neighborhood $\mathcal{N}_{k+1}$ is $\widetilde{X}_E$, without considering the solutions of $\widetilde{X}_E$ that could already be Pareto local optimal for $\mathcal{N}_{k+1}$.

The method stops when a Pareto local optimum set has been found, according to all the neighborhood structures considered.

Algorithm 1. 2PPLS with VNS

```
Parameters ↓: an initial population $P_0$, neighborhood functions $\mathcal{N}_k(x)$, $k_{\min}$, $k_{\max}$.
Parameters ↑: an approximation $\widetilde{X}_E$ of the efficient set $X_E$.

--| Initialization of $\widetilde{X}_E$ and a population $P$ with the initial population $P_0$
$\widetilde{X}_E \leftarrow P_0$
$P \leftarrow P_0$
--| Initialization of an auxiliary population $P_a$
$P_a \leftarrow \varnothing$
--| Initialization of the neighborhood structure
$k \leftarrow k_{\min}$
repeat
  while $P \neq \varnothing$ do
    --| Generation of all neighbors $p'$ of each solution $p \in P$
    for all $p \in P$ do
      for all $p' \in \mathcal{N}_k(p)$ do
        if $f(p) \npreceq_P f(p')$ then
          AddSolution($\widetilde{X}_E \updownarrow$, $p' \downarrow$, $f(p') \downarrow$, $Added \uparrow$)
          if $Added = true$ then
            AddSolution($P_a \updownarrow$, $p' \downarrow$, $f(p') \downarrow$)
  if $P_a \neq \varnothing$ then
    --| $P$ is composed of the new potentially efficient solutions
    $P \leftarrow P_a$
    --| Reinitialization of $P_a$
    $P_a \leftarrow \varnothing$
    --| We start again with the smallest neighborhood structure
    $k \leftarrow k_{\min}$
  else
    --| We use a larger neighborhood structure
    $k \leftarrow k + 1$
    --| We use as population the solutions of $X_E$ that are not Pareto local optimum
    for $\mathcal{N}_k(x)$
    $P \leftarrow \widetilde{X}_E \backslash \{x \in \widetilde{X}_E \,|\, x$ Pareto local optimum for $\mathcal{N}_k(x)\}$
until $k > k_{\max}$
```

4 Results

We will mainly focus the presentation of the results on showing that the use of VLSNS gives a competitive method. We will not have enough space to show all the results of the parameters' analysis of our methods. We will however point out the most important conclusions obtained.

To compare the quality of the approximations generated, we use four indicators: the hypervolume $\mathcal{H}$ (to be maximized) [35], the average distance D_1 and maximal distance D_2 (to be minimized) [10] between the points of Y_N and the points of $\widetilde{Y}_N$, by using the Euclidean distance, and the proportion P_{Y_N} (to be maximized) of non-dominated points generated.

The last three indicators can be used since we have mainly considered instances of the MOMKP and the MOSCP for which we were able to generate exactly Y_N with the ϵ-constraint method [24].

We will only present some results for biobjective instances. The computer used for the experiments is a Pentium IV with 3 GHz CPUs and 512 MB of RAM. Twenty runs of the algorithms are performed for each instance. To compute the ratios R_1, R_2, R_3 and R_4 used in the VLSNS, random generations of weight sets are considered.

4.1 The Multiobjective Multidimensional Knapsack Problem

The MOMKP is one of the most studied MOCO problem. This problem has been used by many different groups of authors in order to test the performances of new multiobjective metaheuristics. A survey of this problem can be found in [27].

As many authors previously did, we use the instances of Zitzler and Thiele [37] with 250, 500 or 750 items, two objectives and two constraints (the instances are called 250-2, 500-2 and 750-2).

Adaptation of 2PPLS. To adapt 2PPLS to the resolution of the MOMKP, we first need to define how the initial population is generated.

As we did not find any efficient published implementations of heuristics solving the single-objective MKP, we have implemented a simple greedy heuristic. To create a new solution, the items are added to the knapsack one by one. At each iteration, the item s that maximizes the ratio (R_2) (see section 2.1) is selected.

The greedy procedure is run with 100 weight sets uniformly generated in $[0, 1]$. We have studied the influence of the number of weight sets, but this number has a low influence on the quality of the results, given the quality of the neighborhood used in PLS.

Influence of the size of the neighborhood. We show now some results about the influence of the size of the neighborhood k. For the MOMKP, no VNS is used.

In Figure 1, we show the evolution of P_{Y_N} and the running time according to k for the 500-2 instance. We vary the values of k from 4 to 20. We use three different numbers of iterations for MEMOTS (the method used to solve the residual problems defined in the VLSNS): $N = 100$, $N = 200$ and $N = 400$. We see that for small values of k, P_{Y_N} is more or less equal no matter the number of iterations. From k equal to 10, it is clear that we obtain better results if N is higher. On the other hand, the running time is bigger when N is higher, but still evolves more or less linearly according to k. An interesting behavior is pointed out by the figure showing the evolution of P_{Y_N} according to k. From k equal to about 16 and for a number of iterations N equal to 100 or 200, there is a deterioration of P_{Y_N} while the running time is increasing. It means that the number of iterations performed in MEMOTS is not high enough to solve the

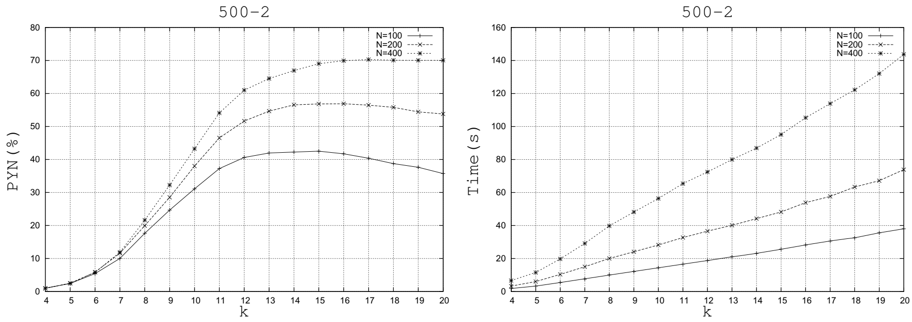


Fig. 1. Influence of k for the MOMKP

residual problems, and that therefore the quality of the approximations obtained for the residual problems is not good enough to improve P_{Y_N}. Fixing good values for k and N should be thus carefully done since these two values have to be increased at the same time if we want to improve the quality of the results.

Comparison to state-of-the-art results. We have obtained the following results for the 250-2, 500-2, 750-2 instances: SPEA [37] (30 runs); SPEA2 [36] (30 runs, but only for the 750-2 instance); MOGLS00 [19] (20 runs); MOGLS04 [19] (20 runs, different than MOGLS00 since obtained with the MOMHLib++ library [18]); PMA [22] (20 runs); IMMOGLS [17] (20 runs); MOGTS [7] (1 run); GRASP [34] (1 run); MOTGA [4] (20 runs); PATH-RELINKING [8] (30 runs); GPLS [2] (30 runs); mGPLS [3] (30 runs); iGPLS [3] (30 runs). These results have been obtained from web sites or directly from the different authors. We see that we have obtained quite a lot of results. It is only a pity that Gomes da Silva *et al.* [11] did not send us their results.

Thanks to these results, we have generated a reference set, called ALL, formed by merging the potentially non-dominated points obtained by all the runs of all algorithms, which gives a very high quality set.

However, we show that is possible to obtain better results than this set, for the indicators considered in this paper, in reasonable times, with the VLSNS integrated into 2PPLS. We have carefully selected the parameters such that we obtain better or equal results than the reference set ALL for all indicators. The parameters are the following:

- 250-2: $k = 9$ and $N = 200$.
- 500-2: $k = 15$ and $N = 100$.
- 750-2: $k = 9$ and $N = 100$.

The results for 2PPLS are given in Table 1. $|PE|$ gives the number of potentially efficient solutions generated. We see that is possible to obtain better or equal values for all indicators, in very reasonable running times: 7s for 250-2, 23s for 500-2 and 18s for 750-2.

Table 1. Comparison between 2PPLS and ALL based on the indicators

Instance	Algorithm	$\mathcal{H}(10^7)$	D_1	D_2	$\|PE\|$	P_{Y_N} (%)	Time(s)
250-2	2PPLS	**9.8690**	**0.029**	**2.680**	482.10	**68.05**	7.27
	ALL	**9.8690**	0.069	2.838	376.00	31.87	/
500-2	2PPLS	**40.7873**	**0.025**	**1.976**	1131.00	**42.85**	23.43
	ALL	40.7850	0.081	2.045	688.00	5.51	/
750-2	2PPLS	**89.3485**	**0.076**	**1.494**	1558.90	**4.15**	17.52
	ALL	89.3449	0.092	**1.494**	996.00	0.99	/

We compare now MEMOTS and 2PPLS, for different running times. In [26], we have showed that MEMOTS was a performing method since this method gives better values than MOGLS [19] and PMA [22] for different indicators.

The results are presented in Figure 2, for the 250-2 and 750-2 instances, where the evolutions of D_1 and P_{Y_N} according to the running time are showed. The running time of 2PPLS is controlled by k and N, while the running of MEMOTS is controlled by the number of iterations performed.

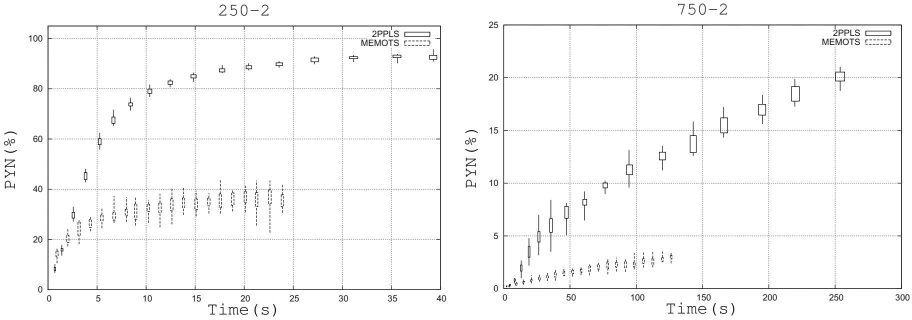


Fig. 2. Comparison of MEMOTS and 2PPLS: evolution of D_1 and P_{Y_N} according to the running time

We see that except with small running times, the results obtained with 2PPLS are better than with MEMOTS. With 2PPLS, we can generate, for the 250-2 instance, about 90% of the non-dominated points, for the 500-2 instance, about 70% and for the 750-2 instance, about 20%, in reasonable running times. The running times are remarkable since, for example, Mavrotas *et al.* [29] can generate for the 250-2 instance, 81% of Y_N, but in about 30 minutes, while we can attain this result in about 15s. Also, for this same instance, they need 21 hours to generate 93% of Y_N, while we only need 42 seconds. We are thus 1800 times faster!

4.2 The Multiobjective Set Covering Problem

The MOSCP has not received as much attention as the MOMKP. To our knowledge, only two groups of authors have tackled this problem. Jaszkiewicz was the first one, in 2001 [20], with the adaptation of the Pareto memetic algorithm (PMA). In 2006, Prins *et al.* [32] have also tackled this problem, by using a two-phase heuristic method (called TPM) using primal-dual Lagrangian relaxations to solve different single-objective SCPs.

We use the same instances than those authors considered, from the size 600x60 (600 sets, 60 items) to the size 1000x200. Seven instances have been considered, and for each size instance, four different kinds of objectives (a, b, c and d) are defined [14]. More informations about these instances can be found in [20,32].

Adaptation of 2PPLS. As initial population, we use a good approximation of the supported efficient solutions. These solutions can be generated by resolution of weighted sum single-objective problems obtained by applying a linear aggregation of the objectives. To generate the different weight sets, we have used the dichotomic method of Aneja and Nair [5]. Depending on the difficulty of the instances, we have used an exact solver (the free MILP solver lp_solve) or a heuristic (the meta-raps heuristic developed by Lan *et al.* [23] whose the code has been published) to solve the single-objective SCPs.

Influence of the size of the neighborhood. We show now some results about the influence of the size of the neighborhood k and the length L of the lists $\mathcal{L}$ and $\mathcal{I}$. For the MOSCP, VNS is used.

In Figure 3, we show the evolution of P_{Y_N} and the running time according to L for the 61a instance (600x60), for different values of k. We vary the values of L from 1 to 9 and k from 1 to 4. We see that, of course, increasing the values of L or k allows to obtain better quality results. The best improvement is when k is moved from 1 to 2 for values of L superior to 5. On the other hand, using $k = 3$ or $k = 4$ in place of $k = 2$ does not give impressive improvements, whatever the value of L.

Concerning the running time, as we use an exact enumeration algorithm to solve the residual problems, we see that its evolution is exponential. Using $k = 3$ or $k = 4$ with $L \geq 8$ becomes very time-consuming.

Therefore, for the comparison to state-of-the-art results, we will use $k = 2$ and $L = 9$.

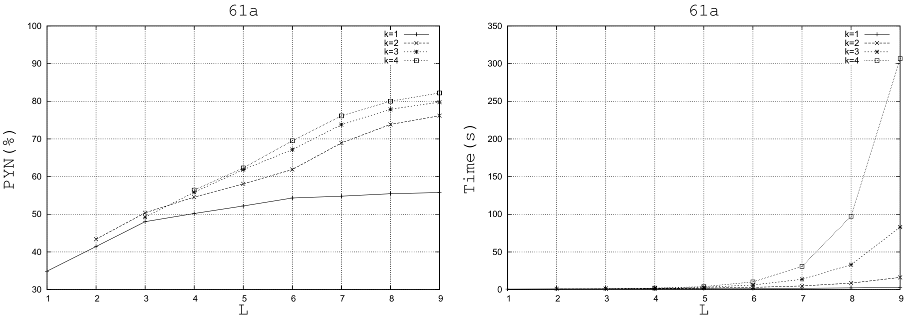


Fig. 3. Influence of k and L for the MOSCP

Comparison to state-of-the-art results. In Table 2, we compare the results obtained with 2PPLS to the results of TPM and PMA. We only show the results for the instance 81 of size 800x80, for the four different kinds of objectives (a, b, c and d) . We see that we obtain better results for the indicators considered. For these instances, we always obtain a value of P_{Y_N} superior to 50%, in less than 41s.

Table 2. Comparison between 2PPLS, TPM and PMA based on the indicators

Instance	Algorithm	$\mathcal{H}(10^5)$	D_1	D_2	P_{Y_N} (%)	$\|PE\|$	Time(s)
81a	2PPLS	**1804.7490**	**0.094**	**1.164**	**58.87**	351.65	31.79
	TPM	1787.6098	1.957	9.729	10.61	64.00	/
	PMA	1777.7066	1.253	3.792	0.61	173.80	/
81b	2PPLS	**2295.7315**	**0.141**	**1.176**	**56.20**	272.25	30.97
	TPM	2277.4616	1.690	6.398	11.30	55.00	/
	PMA	2267.9781	1.197	6.756	3.64	138.10	/
81c	2PPLS	**114.3475**	**0.711**	**3.820**	**71.43**	11.55	19.5
	TPM	114.0509	3.596	15.592	35.71	9.00	/
	PMA	110.0607	21.113	61.595	0.00	1.50	/
81d	2PPLS	**169.5468**	**0.073**	**0.875**	**91.67**	11.00	40.58
	TPM	169.3999	2.827	23.631	75.00	10.00	/
	PMA	160.0092	10.640	42.617	0.00	6.60	/

We have obtained similar results for all the other instances, except for the instances 62c and 62d for which TPM obtains slightly better values for some indicators. For example, for the bigger instances 201a and 201b (1000x200), for which we were not able to generate Y_N, we have obtained that, in average, more than 80% of the points generated by 2PPLS dominate the points generated by PMA.

The comparison of the average running times for solving the different instances is given in Table 3. We see that the running times of 2PPLS are less or equal than the running times of TPM and PMA (coming however from slower computers).

Table 3. Comparison of the average running times of 2PPLS, TPM and PMA (in seconds)

Instance	Dimension	2PPLS	TPM	PMA
61	60 x 600	20.41	20.27	132.4
62	60 x 600	9.54	20.35	98.8
81	80 x 800	30.71	30.22	165.9
82	80 x 800	16.41	30.27	148.4
101	100 x 1000	26.48	50.10	311.7
102	100 x 1000	23.75	50.41	282.1
201	200 x 1000	60.36	70.81	686.8

5 Conclusion

We have shown through this paper the effectiveness of approaches based on VLSNS to solve MOCO problems. The integration of VLSNS into 2PPLS allowed to obtain new state-of-the-art results for two MOCO problems: the MOMKP and MOSCP. However, these results are only for biobjective instances. We have performed some tests for three-objective instances of the MOMKP. The results obtained were of high quality too, but with a very high running time (more than 7h). Indeed, the number of potentially non-dominated points generated was very high (more than 60000 points for an instance with only 250 items). Therefore, in order to make this new approach more practical, the integration of the preferences of the decision maker into the Pareto dominance will be necessary to tackle instances with more than two objectives.

References

1. Ahuja, R.K., Ergun, Ö., Orlin, J.B., Punnen, A.P.: A survey of very large-scale neighborhood search techniques. Discrete Appl. Math. 123(1-3), 75–102 (2002)
2. Alsheddy, A., Tsang, E.P.K.: Guided Pareto local search and its application to the 0/1 multi-objective knapsack problems. In: Proceedings of the Eighth Metaheuristic International Conference (MIC 2009), Hamburg (2009)
3. Alsheddy, A., Tsang, E.P.K.: Guided Pareto local search based frameworks for Pareto optimization. In: Proceedings of the WCCCI IEEE World Congress on Computational Intelligence, Barcelona (2010)
4. Alves, M.J., Almeida, M.: MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. Computers & Operations Research 34, 3458–3470 (2007)
5. Aneja, Y.P., Nair, K.P.K.: Bicriteria transportation problem. Management Science 25, 73–78 (1979)
6. Angel, E., Bampis, E., Gourvés, L.: A dynasearch neighborhood for the bicriteria traveling salesman problem. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds.) Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, vol. 535, pp. 153–176. Springer, Berlin (2004)
7. Barichard, V., Hao, J.K.: An empirical study of tabu search for the MOKP. In: Proceedings of the First International Workshop on Heuristics, China. Series of Information & Management Sciences, vol. 4, pp. 47–56 (2002)
8. Beausoleil, R.P., Baldoquin, G., Montejo, R.A.: Multi-start and path relinking methods to deal with multiobjective knapsack problems. Annals of Operations Research 157, 105–133 (2008)
9. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002)
10. Czyzak, P., Jaszkiewicz, A.: Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis 7, 34–47 (1998)
11. Gomes da Silva, C., Clímaco, J., Figueira, J.R.: Scatter search method for the bi-criteria multi-dimensional {0,1}-knapsack problem using surrogate relaxation. Journal of Mathematical Modelling and Algorithms 3(3), 183–208 (2004)

12. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, New York (2001)
13. Ehrgott, M., Gandibleux, X.: Multiobjective combinatorial optimization. In: Ehrgott, M., Gandibleux, X. (eds.) Multiple Criteria Optimization – State of the Art Annotated Bibliographic Surveys, vol. 52, pp. 369–444. Kluwer Academic Publishers, Boston (2002)
14. Gandibleux, X., Vancoppenolle, D., Tuyttens, D.: A first making use of GRASP for solving MOCO problems. In: 14th International Conference in Multiple Criteria Decision-Making, Charlottesville (1998)
15. Glover, F., Kochenberger, G.: Handbook of Metaheuristics. Kluwer, Boston (2003)
16. Hansen, P., Mladenovic, N.: Variable neighborhood search: Principles and applications. European Journal of Operational Research 130(3), 449–467 (2001)
17. Ishibuchi, H., Murada, T.: A multi-objective genetic local search algorithm and its application to flow shop scheduling. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews 28(3), 392–403 (1998)
18. Jaszkiewicz, A.: Experiments done with the MOMHLIB: Technical report, Institute of Computing Science, Poznań University of Technology (2000), http://www-idss.cs.put.poznan.pl/jaszkiewicz/momhlib/
19. Jaszkiewicz, A.: On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem—A Comparative Experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznań University of Technology, Poznań, Poland (July 2000)
20. Jaszkiewicz, A.: A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the Pareto memetic algorithm. Technical Report RA-003/01, Institute of Computing Science, Poznań University of Technology, Poznań, Poland (2001)
21. Jaszkiewicz, A.: On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem—A Comparative Experiment. IEEE Transactions on Evolutionary Computation 6(4), 402–412 (2002)
22. Jaszkiewicz, A.: On the Computational Efficiency of Multiple Objective Metaheuristics. The Knapsack Problem Case Study. European Journal of Operational Research 158(2), 418–433 (2004)
23. Lan, G., DePuy, G.W., Whitehouse, G.E.: An effective and simple heuristic for the set covering problem. European Journal of Operational Research 176(3), 1387–1403 (2007)
24. Laumanns, M., Thiele, L., Zitzler, E.: An adaptative scheme to generate the Pareto front based on the epsilon-constraint method. Technical Report 199, Technischer Bericht, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology, ETH (2004)
25. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Operations Research 21, 498–516 (1973)
26. Lust, T., Teghem, J.: Memots: a memetic algorithm integrating tabu search for combinatorial multiobjective optimization. RAIRO: Operations Research 42(1), 3–33 (2008)
27. Lust, T., Teghem, J.: The multiobjective multidimensionnal knapsack problem: a survey and a new approach. Technical Report arXiv:1007.4063v1, arXiv (2010)
28. Lust, T., Teghem, J.: Two-phase Pareto local search for the biobjective traveling salesman problem. Journal of Heuristics 16(3), 475–510 (2010)
29. Mavrotas, G., Figueira, J.R., Florios, K.: Solving the bi-objective multi-dimensional knapsack problem exploiting the concept of core. Applied Mathematics and Computation 215(7), 2502–2514 (2009)

30. Michalewicz, Z., Arabas, J.: Genetic algorithms for the 0/1 knapsack problem. In: Raś, Z.W., Zemankova, M. (eds.) ISMIS 1994. LNCS, vol. 869, pp. 134–143. Springer, Heidelberg (1994)
31. Paquete, L., Chiarandini, M., Stützle, T.: Pareto Local Optimum Sets in the Biobjective Traveling Salesman Problem: An Experimental Study. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds.) Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, vol. 535, pp. 177–199. Springer, Berlin (2004)
32. Prins, C., Prodhon, C., Wolfler Calvo, R.: Two-phase method and lagrangian relaxation to solve the bi-objective set covering problem. Annals OR 147(1), 23–41 (2006)
33. Ulungu, E.L., Teghem, J.: Multiobjective combinatorial optimization problems: A survey. Journal of Multi-Criteria Decision Analysis 3, 83–104 (1994)
34. Vianna, D.S., Arroyo, J.E.C.: A GRASP algorithm for the multi-objective knapsack problem. In: QEST 2004: Proceedings of the The Quantitative Evaluation of Systems, First International Conference, pp. 69–75. IEEE Computer Society, Washington, DC (2004)
35. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (November 1999)
36. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (May 2001)
37. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

Bilevel Multi-objective Optimization Problem Solving Using Progressively Interactive EMO

Ankur Sinha

Department of Business Technology, Aalto University School of Economics
PO Box 21210, FIN-00076 Aalto, Helsinki, Finland
Ankur.Sinha@aalto.fi

Abstract. Bilevel multi-objective optimization problems are known to be highly complex optimization tasks which require every feasible upper-level solution to satisfy optimality of a lower-level optimization problem. Multi-objective bilevel problems are commonly found in practice and high computation cost needed to solve such problems motivates to use multi-criterion decision making ideas to efficiently handle such problems. Multi-objective bilevel problems have been previously handled using an evolutionary multi-objective optimization (EMO) algorithm where the entire Pareto set is produced. In order to save the computational expense, a progressively interactive EMO for bilevel problems has been presented where preference information from the decision maker at the upper level of the bilevel problem is used to guide the algorithm towards the most preferred solution (a single solution point). The procedure has been evaluated on a set of five DS test problems suggested by Deb and Sinha. A comparison for the number of function evaluations has been done with a recently suggested Hybrid Bilevel Evolutionary Multi-objective Optimization algorithm which produces the entire upper level Pareto-front for a bilevel problem.

Keywords: Genetic algorithms, evolutionary algorithms, bilevel optimization, multi-objective optimization, evolutionary programming, multi-criteria decision making, hybrid evolutionary algorithms, sequential quadratic programming.

1 Introduction

Bilevel programming problems are often found in practice ([25]) where the feasibility of an upper level solution is decided by a lower level optimization problem. The qualification for an upper level solution to be feasible is that it should be an optimal candidate from a lower level optimization problem. This requirement consequentially makes a bilevel problem very difficult to handle. Multiple objectives at both the levels of a bilevel problem further adds to the complexity. Because of difficulty in searching and defining optimal solutions for bilevel multi-objective optimization problems ([11]), not many solution methodologies to such problems have been explored. One of the recent advances made in this direction is by Deb and Sinha ([9]) where the entire Pareto set at the upper level of the bilevel multi-objective problem is explored. The method, though successful in handling complex bilevel multi-objective test problems, is computationally expensive and requires high function evaluations, particularly at the lower level. High

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 269–284, 2011.

computational expense associated to such problems provides a motivation to explore a different solution methodology.

Concepts from a Progressively Interactive Evolutionary Multi-objective Optimization algorithm (PI-EMO-VF) ([10]) has been integrated with the Hybrid Bilevel Evolutionary Multi-objective Optimization algorithm (HBLEMO) ([9]) in this paper. In the suggested methodology, preference information from the decision maker at the upper level is used to direct the search towards the most preferred solution. Incorporating preferences from the decision maker in the optimization run makes the search process more efficient in terms of function evaluations as well as accuracy. The integrated methodology proposed in this paper, interacts with the decision maker after every few generations of an evolutionary algorithm and is different from an a posteriori approach, as it explores only the most preferred point. An a posteriori approach like the HBLEMO and other evolutionary multi-objective optimization algorithms ([5]; [26]) produce the entire efficient frontier as the final solution and then a decision maker is asked to pick up the most preferred point. However, an a posteriori approach is not a viable methodology for problems which are computationally expensive and/or involve high number of objectives (more than three) where EMOs tend to suffer in convergence as well as maintaining diversity.

In this paper, the bilevel multi-objective problem has been described initially and then the integrated procedure, Progressively Interactive Hybrid Bilevel Evolutionary Multi-objective Optimization (PI-HBLEMO) algorithm, has been discussed. The performance of the PI-HBLEMO algorithm has been shown on a set of five DS test problems ([9]; [6]) and a comparison for the savings in computational cost has been done with a posteriori HBLEMO approach.

2 Recent Studies

In the context of bilevel single-objective optimization problems a number of studies exist, including some useful reviews ([3]; [21]), test problem generators ([2]), and some evolutionary algorithm (EA) studies ([18]; [17]; [24]; [16]; [23]). Stackelberg games ([13]; [22]), which have been widely studied, are also in principle similar to a single-objective bilevel problem. However, not many studies can be found in case of bilevel multi-objective optimization problems. The bilevel multi-objective problems have not received much attention, either from the classical researchers or from the researchers in the evolutionary community.

Eichfelder ([12]) worked on a classical approach on handling multi-objective bilevel problems, but the nature of the approach made it limited to handle only problems with few decision variables. Halter et al. ([15]) used a particle swarm optimization (PSO) procedure at both the levels of the bilevel multi-objective problem but the application problems they used had linearity in the lower level. A specialized linear multi-objective PSO algorithm was used at the lower level, and a nested strategy was utilized at the upper level.

Recently, Deb and Sinha have proposed a Hybrid Bilevel Evolutionary Multi-objective Optimization algorithm (HBLEMO) ([9]) using NSGA-II to solve both level problems in a synchronous manner. Former versions of the HBLEMO algorithm can be found in the conference publications (6; 8; 19; 7). The work in this paper extends the HBLEMO algorithm by allowing the decision maker to interact with the algorithm.

3 Multi-objective Bilevel Optimization Problems

In a multi-objective bilevel optimization problem there are two levels of multi-objective optimization tasks. A solution is considered feasible at the upper level only if it is a Pareto-optimal member to a lower level optimization problem (9). A generic multi-objective bilevel optimization problem can be described as follows. In the formulation there are M number of objectives at the upper level and m number of objectives at the lower level:

$$\begin{aligned} \text{Minimize}_{(\mathbf{x}_u,\mathbf{x}_l)} \ & \mathbf{F}(\mathbf{x}) = (F_1(\mathbf{x}), \ldots, F_M(\mathbf{x})), \\ \text{subject to } & \mathbf{x}_l \in \text{argmin}_{(\mathbf{x}_l)} \left\{ \mathbf{f}(\mathbf{x}){=}(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \,\middle|\, \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0} \right\}, \\ & \mathbf{G}(\mathbf{x}) \geq \mathbf{0}, \mathbf{H}(\mathbf{x}) = \mathbf{0}, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, \ldots, n. \end{aligned} \tag{1}$$

In the above formulation, $F_1(\mathbf{x}), \ldots, F_M(\mathbf{x})$ are upper level objective functions which are M in number and $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ are lower level objective functions which are m in number. The constraint functions $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ determine the feasible space for the lower level problem. The decision vector, $\mathbf{x}$, contains the variables to be optimized at the upper level. It is composed of two smaller vectors $\mathbf{x}_u$ and $\mathbf{x}_l$, such that $\mathbf{x} = (\mathbf{x}_u, \mathbf{x}_l)$. While solving the lower level problem, it is important to note that the lower level problem is optimized with respect to the variables $\mathbf{x}_l$ and the variables $\mathbf{x}_u$ act as fixed parameters for the problem. Therefore, the Pareto-optimal solution set to the lower level problem can be represented as $\mathbf{x}_l^*(\mathbf{x}_u)$. This representation means that the upper level variables $\mathbf{x}_u$, act as a parameter to the lower level problem and hence the lower level optimal solutions $\mathbf{x}_l^*$ are a function of the upper level vector $\mathbf{x}_u$. The functions $\mathbf{G}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ define the feasibility of a solution at the upper level along with the Pareto-optimality condition to the lower level problem.

4 Progressively Interactive Hybrid Bilevel Evolutionary Multi-objective Optimization Algorithm (PI-HBLEMO)

In this section, the changes made to the Hybrid Bilevel Evolutionary Multi-objective Optimization (HBLEMO) (9) algorithm have been stated. The major change made to the HBLEMO algorithm is in the domination criteria. The other change which has been made is in the termination criteria.

The Progressively Interactive EMO using Value Function (PI-EMO-VF) (10) is a generic procedure which can be integrated with any Evolutionary Multi-objective Optimization (EMO) algorithm. In this section we integrate the procedure at the upper level execution of the HBLEMO algorithm.

After every τ upper level generations of the HBLEMO algorithm, the decision-maker is provided with η (≥ 2) well-sparse non-dominated solutions from the upper level set of non-dominated points. The decision-maker is expected to provide a complete or partial preference information about superiority of one solution over the other, or indifference towards the two solutions. In an ideal situation, the DM can provide a complete ranking (from best to worst) of these solutions, but partial preference information is also allowed. With the given preference information, a strictly increasing polynomial value function is constructed. The value function construction procedure involves solving another single-objective optimization problem. Till the next τ upper level generations, the constructed value function is used to direct the search towards additional preferred solutions.

The termination condition used in the HBLEMO algorithm is based on hypervolume. In the modified PI-HBLEMO algorithm the search is for the most preferred point and not for a pareto optimal front, therefore, the hypervolume based termination criteria can no longer be used. The hypervolume based termination criteria at the upper level has been replaced with a criteria based on distance of an improved solution from the best solutions in the previous generations.

In the following, we specify the steps required to blend the HBLEMO algorithm within the PI-EMO-VF framework and then discuss the termination criteria.

1. Set a counter $t = 0$. Execute the HBLEMO algorithm with the usual definition of dominance (14) at the upper level for τ generations. Increment the value of t by one after each generation.
2. If t is perfectly divisible by τ, then use the k-mean clustering approach ((4; 26)) to choose η diversified points from the non-dominated solutions in the archive; otherwise, proceed to Step 5.
3. Elicitate the preferences of the decision-maker on the chosen η points. Construct a value function $V(\mathbf{f})$, emulating the decision maker preferences, from the information. The value function is constructed by solving an optimization problem (VFOP), described in Section 4.1. If a feasible value function is not found which satisfies all DM's preferences then proceed to Step 5 and use the usual domination principle in HBLEMO operators.
4. Check for termination. The termination check (described in Section 4.2) is based on the distance of the current best solution from the previous best solutions and requires a parameter ϵ_u. If the algorithm terminates, the current best point is chosen as the final solution.
5. An offspring population at the upper level is produced from the parent population at the upper level using a modified domination principle (discussed in Section 4.3) and HBLEMO algorithm's search operators.
6. The parent and the offspring populations are used to create a new parent population for the next generation using the modified domination based on the current value function and other HBLEMO algorithm's operators. The iteration counter is incremented as $t \leftarrow t + 1$ and the algorithm proceeds to Step 2.

The parameters used in the PI-HBLEMO algorithm are τ, η and ϵ_u.

4.1 Step 3: Elicitation of Preference Information and Construction of a Polynomial Value Function

Whenever a DM call is made, a set of η points are presented to the decision maker (DM). The preference information from the decision maker is accepted in the form of pairwise comparisons for each pair in the set of η points. A pairwise comparison of a give pair could lead to three possibilities, the first being that one solution is preferred over the other, the second being that the decision maker is indifferent to both the solutions and the third being that the two solutions are incomparable. Based on such preference information from a decision maker, for a given pair (i, j), if i-th point is preferred over j-th point, then $P_i \succ P_j$, if the decision maker is indifferent to the two solutions then it establishes that $P_i \equiv P_j$. There can be situations such that the decision maker finds a given pair of points as incomparable and in such a case the incomparable points are dropped from the list of η points. If the decision maker is not able to provide preference information for any of the given solution points then algorithm moves back to the previous population where the decision maker was able to take a decisive action, and uses the usual domination instead of modified domination principle to proceed the search process. But such a scenario where no preference is established by a decision maker is rare, and it is likely to have at least one point which is better than another point. Once preference information is available, the task is to construct a polynomial value function which satisfies the preference statements of the decision maker.

Polynomial Value Function for Two Objectives

A polynomial value function is constructed based on the preference information provided by the decision maker. The parameters of the polynomial value function are optimally adjusted such that the preference statements of the decision maker are satisfied. We describe the procedure for two objectives as all the test problems considered in this paper have two objectives. The value function procedure described below is valid for a maximization problem therefore we convert the test problems used in this paper into a maximization problem while implementing the value function procedure. However, while reporting the results for the test problems they are converted back to minimization problems.

$$\begin{array}{lll} & V(F_1, F_2) = (F_1 + k_1 F_2 + l_1)(F_2 + k_2 F_1 + l_2), & \\ \text{where } F_1, F_2 & \text{are the objective values} & \\ \text{and } \quad k_1, k_2, l_1, l_2 & \text{are the value function parameters} & \end{array} \tag{2}$$

The description of the two objective value function has been taken from ([10]). In the above equations it can been seen that the value function V, for two objectives, is represented as a product of two linear functions $S_1 : \mathbf{R}^2 \rightarrow \mathbf{R}$ and $S_2 : \mathbf{R}^2 \rightarrow \mathbf{R}$. [1] The parameters in this value function which are required to be determined optimally from the preference statements of the decision maker are k_1, k_2, l_1 and l_2. Following is the value function optimization problem (VFOP) which should be solved with the value function parameters (k_1, k_2, l_1 and l_2) as variables. The optimal solution to the VFOP assigns optimal values to the value function parameters. The above problem is a simple single objective optimization problem which can be solved using any single

[1] A generalized version of the polynomial value function can be found in ([20]).

objective optimizer. In this paper the problem has been solved using a sequential quadratic programming (SQP) procedure from the KNITRO ([1]) software.

$$
\begin{aligned}
\text{Maximize } & \epsilon, \\
\text{subject to } & V \text{ is non-negative at every point } P_i, \\
& V \text{ is strictly increasing at every point } P_i, \\
& V(P_i) - V(P_j) \geq \epsilon, \quad \text{for all } (i,j) \text{ pairs} \\
& \qquad\qquad \text{satisfying } P_i \succ P_j, \\
& |V(P_i) - V(P_j)| \leq \delta_V, \quad \text{for all } (i,j) \text{ pairs} \\
& \qquad\qquad \text{satisfying } P_i \equiv P_j.
\end{aligned}
\tag{3}
$$

The above optimization problem adjusts the value function parameters in such a way that the minimum difference in the value function values for the ordered pairs of points is maximum.

4.2 Termination Criterion

Distance of the current best point is computed from the best points in the previous generations. In the simulations performed, the distance is computed from the current best point to the best points in the previous 10 generations and if each of the computed distances $\delta_u(i), i \in \{1, 2, \ldots, 10\}$ is found to be less than ϵ_u then the algorithm is terminated. A value of $\epsilon_u = 0.1$ has been chosen for the simulations done in this paper.

4.3 Modified Domination Principle

In this sub-section we define the modified domination principle proposed in ([10]). The value function V is used to modify the usual domination principle so that more focussed search can be performed in the region of interest to the decision maker. Let $V(F_1, F_2)$ be the value function for a two objective case. The parameters for this value function are optimally determined from the VFOP. For the given η points, the value function assigns a value to each point. Let the values be $V_1, V_2, \ldots, V_\eta$ in the descending order. Now any two feasible solutions ($\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$) can be compared with their objective function values by using the following modified domination criteria:

1. If both points have a value function value *less* than V_2, then the two points are compared based on the usual dominance principle.
2. If both points have a value function value *more* than V_2, then the two points are compared based on the usual dominance principle.
3. If one point has a value function value more than V_2 and the other point has a value function value less than V_2, then the former dominates the latter.

The modified domination principle has been explained through Figure 1 which illustrates regions dominated by two points A and B. Let us consider that the second best point from a given set of η points has a value V_2. The function $V(F) = V_2$ represents a contour which has been shown by a curved line [2]. The first point A has a value V_A

[2] The reason for using the contour corresponding to the second best point can be found in ([10]).

which is smaller than V_2 and the region dominated by A is shaded in the figure. The region dominated by A is identical to what can be obtained using the usual domination principle. The second point B has a value V_B which is larger than V_2, and, the region dominated by this point is once again shaded. It can be observed that this point no longer follows the usual domination principle. In addition to usual region of dominance this point dominates all the points having a smaller value function value than V_2.

The above modified domination principle can easily be extended for handling constraints as in ([5]). When two solutions under consideration for a dominance check are feasible, then the above modified domination principle should be used. If one solution is feasible and the other is infeasible, then the feasible solution is considered as dominating the other. If both the solutions are found to be infeasible then the one with smaller overall feasibility violation (sum of all constraint violations) is considered to be dominating the other solution.

5 Parameter Setting

In the next section, results of the PI-HBLEMO and the HBLEMO procedure on the set of DS test problems (([9])) have been presented. In all simulations, we have used the following parameter values for PI-HBLEMO:

1. Number of points given to the DM for preference information: $\eta = 5$.
2. Number of generations between two consecutive DM calls: $\tau = 5$.
3. Termination parameter: $\epsilon_u = 0.1$.
4. Crossover probability and the distribution index for the SBX operator: $p_c = 0.9$ and $\eta_c = 15$.
5. Mutation probability and the distribution index for polynomial mutation: $p_m = 0.1$ and $\eta_m = 20$.
6. Population size: $N = 40$

6 Results

In this section, results have been presented on a set of 5 DS test problems. All the test problems have two objectives at both the levels. A point, $(F_1^{(b)}, F_2^{(b)})$, on the Pareto-front of the upper level is assumed as the most preferred point and then a DM emulated value function is selected which assigns a maximum value to the most preferred point. The value function selected is $V(F_1, F_2) = \frac{1}{1+(F_1-F_1^{(b)})^2+(F_2-F_2^{(b)})^2}$. It is noteworthy that the value function selected to emulate a decision maker is a simple distance function and therefore has circles as indifference curves which is not a true representative of a rational decision maker. A circular indifference curve may lead to assignment of equal values to a pair of points where one dominates the other. For a pair of points it may also lead assignment of higher value to a point dominated by the other. However, only non-dominated set of points are presented to a decision maker, therefore, such discrepancies are avoided and the chosen value function is able to emulate a decision maker by assigning higher value to the point closest to the most preferred point and lower value to others.

The DS test problems are minimization problems and the progressively interactive procedure using value function works only on problems to be maximized. Therefore, the procedure has been executed by converting the test problems into a maximization problem by putting a negative sign before each of the objectives. However, the final results have once again been converted and the solution to the minimization problem has been presented. The upper level and lower level function evaluations have been reported for each of the test problems. A comparison has been made between the HBLEMO algorithm and PI-HBLEMO procedure in the tables 1, 2, 3, 4 and 5. The tables show the savings in function evaluations which could be achieved moving from an a posteriori approach to a progressively interactive approach. The total lower level function evaluations (Total LL FE) and the total upper level function evaluations (Total UL FE) are presented separately. Table 6 shows the accuracy achieved and the number of DM calls required to get close to the most preferred point. The accuracy is the Euclidean distance of the best point achieved by the algorithm from the most preferred point. The most preferred point has been represented on the Pareto-optimal fronts of the test problems.

6.1 Problem DS1

Problem DS1 has been taken from (9). A point on the Pareto-optimal front of the test problem is chosen as the most-preferred point and then the PI-HBLEMO algorithm is executed to obtain a solution close to the most preferred point. This problem has $2K$ variables with K real-valued variables each for lower and upper levels. The complete DS1 problem is given below:

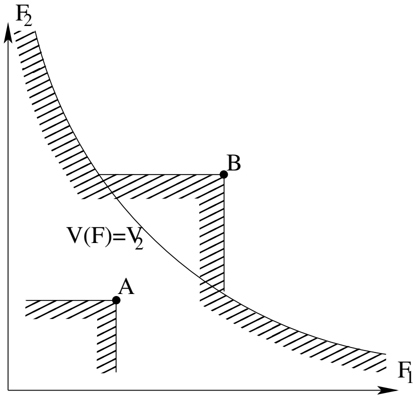


Fig. 1. Dominated regions of two points A and B using the modified definition. Taken from (10).

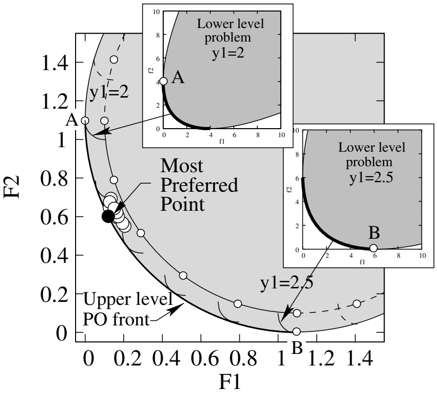


Fig. 2. Pareto-optimal front for problem DS1. Final parent population members have been shown close to the most preferred point.

Figure 2 shows the Pareto-optimal front for the test problem, and the most-preferred solution is marked on the front. The final population members from a particular run of the PI-HBLEMO algorithm are also shown. Table 1 presents the function evaluations required to arrive at the best solution using PI-HBLEMO and also the function evaluations required to achieve an approximated Pareto-frontier using the HBELMO algorithm. The third row in the table presents the ratio of function evaluations using HBLEMO and PI-HBLEMO.

$$\begin{array}{l}\text{Minimize} \quad \mathbf{F}(\mathbf{x},\mathbf{y}) = \\ \left(\begin{array}{l}(1+r-\cos(\alpha\pi y_1)) + \sum_{j=2}^{K}(y_j - \frac{j-1}{2})^2 \\ \quad +\tau\sum_{i=2}^{K}(x_i-y_i)^2 - r\cos\left(\gamma\frac{\pi}{2}\frac{x_1}{y_1}\right) \\ (1+r-\sin(\alpha\pi y_1)) + \sum_{j=2}^{K}(y_j - \frac{j-1}{2})^2 \\ \quad +\tau\sum_{i=2}^{K}(x_i-y_i)^2 - r\sin\left(\gamma\frac{\pi}{2}\frac{x_1}{y_1}\right)\end{array}\right), \\ -K \le x_i \le K, \quad \text{for } i=1,\ldots,K, \\ 1 \le y_1 \le 4, \quad -K \le y_j \le K, \quad j=2,\ldots,K.\end{array} \quad \begin{array}{l}\text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})}\mathbf{f}(\mathbf{x}) = \\ \left\{\left(\begin{array}{l}x_1^2 + \sum_{i=2}^{K}(x_i-y_i)^2 \\ \quad +\sum_{i=2}^{K} 10(1-\cos(\frac{\pi}{K}(x_i-y_i))) \\ \sum_{i=1}^{K}(x_i-y_i)^2 \\ \quad +\sum_{i=2}^{K} 10|\sin(\frac{\pi}{K}(x_i-y_i)|\end{array}\right)\right\},\end{array} \tag{4}$$

For this test problem, $K = 10$ (overall 20 variables), $r = 0.1$, $\alpha = 1$, $\gamma = 1$, and $\tau = 1$ has been used.

Table 1. Total function evaluations for the upper and lower level (21 runs) for DS1

Algo. 1, Algo. 2, Savings	Best		Median		Worst	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE	Total LL FE	Total UL FE
HBLEMO	2,819,770	87,582	3,423,544	91,852	3,829,812	107,659
PI-HBLEMO	329,412	12,509	383,720	12,791	430,273	10,907
$\frac{HBLEMO}{PI-HBLEMO}$	8.56	7.00	8.92	7.18	8.90	9.87

6.2 Problem DS2

Problem DS2 has been taken from (9). A point on the Pareto-optimal front of the test problem is chosen as the most-preferred point and then the PI-HBLEMO algorithm is executed to obtain a solution close to the most preferred point. This problem uses discrete values of y_1 to determine the upper level Pareto-optimal front. The overall problem is given as follows:

$$\begin{aligned} u_1(y_1) &= \begin{cases} \cos(0.2\pi)y_1 + \sin(0.2\pi)\sqrt{|0.02\sin(5\pi y_1)|}, \\ \quad \text{for } 0 \le y_1 \le 1, \\ y_1 - (1-\cos(0.2\pi)), \quad y_1 > 1 \end{cases} \\ u_2(y_1) &= \begin{cases} -\sin(0.2\pi)y_1 + \cos(0.2\pi)\sqrt{|0.02\sin(5\pi y_1)|}, \\ \quad \text{for } 0 \le y_1 \le 1, \\ 0.1(y_1 - 1) - \sin(0.2\pi), \quad \text{for } y_1 > 1. \end{cases} \end{aligned} \tag{5}$$

$$
\begin{aligned}
&\text{Minimize}\quad \mathbf{F}(\mathbf{x},\mathbf{y}) = \\
&\begin{pmatrix}
u_1(y_1) + \sum_{j=2}^{K}\left[y_j^2 + 10(1-\cos(\frac{\pi}{K}y_i))\right] \\
+\tau\sum_{i=2}^{K}(x_i-y_i)^2 - r\cos\left(\gamma\frac{\pi}{2}\frac{x_1}{y_1}\right) \\
u_2(y_1) + \sum_{j=2}^{K}\left[y_j^2 + 10(1-\cos(\frac{\pi}{K}y_i))\right] \\
+\tau\sum_{i=2}^{K}(x_i-y_i)^2 - r\sin\left(\gamma\frac{\pi}{2}\frac{x_1}{y_1}\right)
\end{pmatrix}
\begin{matrix}\text{subject to}\quad (\mathbf{x}) \in \mathbf{f}(\mathbf{x}) = \\ ,\ \text{argmin}_{(\mathbf{x})}\left\{\begin{pmatrix} x_1^2 + \sum_{i=2}^{K}(x_i-y_i)^2 \\ \sum_{i=1}^{K} i(x_i-y_i)^2 \end{pmatrix}\right\},\end{matrix} \\
&-K \le x_i \le K,\quad i = 1,\ldots,K, \\
&0.001 \le y_1 \le K,\quad -K \le y_j \le K,\ j = 2,\ldots,K,
\end{aligned}
\tag{6}
$$

Due to the use of periodic terms in u_1 and u_2 functions, the upper level Pareto-optimal front corresponds to only six discrete values of y_1 (=0.001, 0.2, 0.4, 0.6, 0.8 and 1). $r = 0.25$ has been used.

In this test problem the upper level problem has multi-modalities, thereby causing an algorithm difficulty in finding the upper level Pareto-optimal front. A value of $\tau = -1$ has been used, which introduces a conflict between upper and lower level problems. The results have been produced for 20 variables test problem.

The Pareto-optimal front for this test problem is shown in Figure 3. The most-preferred solution is marked on the Pareto-front along with the final population members obtained from a particular run of the PI-HBLEMO algorithm. Table 2 presents the function evaluations required to arrive at the best solution using PI-HBLEMO. The function evaluations required to achieve an approximated Pareto-frontier using the HBELMO algorithm is also reported.

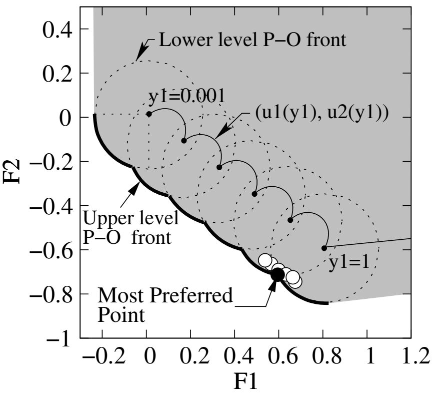


Fig. 3. Pareto-optimal front for problem DS2. Final parent population members have been shown close to the most preferred point.

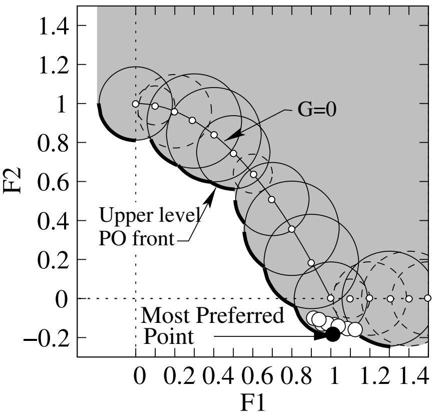


Fig. 4. Pareto-optimal front for problem DS3. Final parent population members have been shown close to the most preferred point.

Table 2. Total function evaluations for the upper and lower level (21 runs) for DS2

Algo. 1, Algo. 2, Savings	Best		Median		Worst	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE	Total LL FE	Total UL FE
HBLEMO	4,796,131	112,563	4,958,593	122,413	5,731,016	144,428
PI-HBLEMO	509,681	14,785	640,857	14,535	811,588	15,967
$\frac{HBLEMO}{PI-HBLEMO}$	9.41	7.61	7.74	8.42	7.06	9.05

6.3 Problem DS3

Problem DS3 has been taken from (9). A point on the Pareto-optimal front of the test problem is chosen as the most-preferred point and then the PI-HBLEMO algorithm is executed to obtain a solution close to the most preferred point. In this test problem, the variable y_1 is considered to be discrete, thereby causing only a few y_1 values to represent the upper level Pareto-optimal front. The overall problem is given below:

$$\begin{array}{l}
\text{Minimize} \quad \mathbf{F}(\mathbf{x},\mathbf{y}) = \\
\begin{pmatrix} y_1 + \sum_{j=3}^{K}(y_j - j/2)^2 + \tau\sum_{i=3}^{K}(x_i - y_i)^2 - R(y_1)\cos(4\tan^{-1}\left(\frac{y_2-x_2}{y_1-x_1}\right) \\ y_2 + \sum_{j=3}^{K}(y_j - j/2)^2 + \tau\sum_{i=3}^{K}(x_i - y_i)^2 - R(y_1)\sin(4\tan^{-1}\left(\frac{y_2-x_2}{y_1-x_1}\right) \end{pmatrix}, \\
\text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \\
\quad \left\{ \mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_1 + \sum_{i=3}^{K}(x_i - y_i)^2 \\ x_2 + \sum_{i=3}^{K}(x_i - y_i)^2 \end{pmatrix} \middle| g_1(\mathbf{x}) = (x_1 - y_1)^2 + (x_2 - y_2)^2 \le r^2 \right\}, \\
G(\mathbf{y}) = y_2 - (1 - y_1^2) \ge 0, \\
-K \le x_i \le K, \quad \text{for } i = 1, \ldots, K, \quad 0 \le y_j \le K, \quad \text{for } j = 1, \ldots, K, \\
y_1 \text{ is a multiple of 0.1.}
\end{array} \qquad (7)$$

Here a periodically changing radius has been used: $R(y_1) = 0.1 + 0.15|\sin(2\pi(y_1 - 0.1)|$ and use $r = 0.2$. For the upper level Pareto-optimal points, $y_i = j/2$ for $j \le 3$. The variables y_1 and y_2 take values satisfying constraint $G(\mathbf{y}) = 0$. For each such combination, variables x_1 and x_2 lie on the third quadrant of a circle of radius r and center at (y_1, y_2) in the $\mathbf{F}$-space. For this test problem, the Pareto-optimal fronts for both lower and upper level lie on constraint boundaries, thereby requiring good constraint handling strategies to solve the problem adequately. $\tau = 1$ has been used for this test problem with 20 number of variables.

Table 3. Total function evaluations for the upper and lower level (21 runs) for DS3

Algo. 1, Algo. 2, Savings	Best		Median		Worst	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE	Total LL FE	Total UL FE
HBLEMO	3,970,411	112,560	4,725,596	118,848	5,265,074	125,438
PI-HBLEMO	475,600	11,412	595,609	16,693	759,040	16,637
$\frac{HBLEMO}{PI-HBLEMO}$	8.35	9.86	7.93	7.12	6.94	7.54

The Pareto-front, most-preferred point and the final population members from a particular run are shown in Figure 4. Table 3 presents the function evaluations required by PI-HBLEMO to produce the final solution and the function evaluations required by HBELMO to produce an approximate Pareto-front.

6.4 Problem DS4

Problem DS4 has been taken from (9). A point on the Pareto-optimal front of the test problem is chosen as the most-preferred point and then the PI-HBLEMO algorithm is executed to obtain a solution close to the most preferred point. This problem has $K+L+1$ variables, which are all real-valued:

$$\begin{array}{ll}
\text{Minimize} \quad \mathbf{F}(\mathbf{x},\mathbf{y}) = & \text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \mathbf{f}(\mathbf{x}) = \\
\begin{pmatrix} (1-x_1)(1+\sum_{j=2}^{K} x_j^2)y_1 \\ x_1(1+\sum_{j=2}^{K} x_j^2)y_1 \end{pmatrix}, & \left\{ \begin{pmatrix} (1-x_1)(1+\sum_{j=K+1}^{K+L} x_j^2)y_1 \\ x_1(1+\sum_{j=K+1}^{K+L} x_j^2)y_1 \end{pmatrix} \right\}, \\
G_1(\mathbf{x}) = (1-x_1)y_1 + \frac{1}{2}x_1 y_1 - 1 \geq 0, & \\
-1 \leq x_1 \leq 1, \quad 1 \leq y_1 \leq 2, & \\
-(K+L) \leq x_i \leq (K+L), \; i = 2, \ldots, (K+L). &
\end{array} \tag{8}$$

The upper level Pareto-optimal front is formed with $x_i = 0$ for all $i = 2, \ldots, (K+L)$ and $x_1 = 2(1-1/y_1)$ and $y_1 \in [1,2]$. By increasing K and L, the problem complexity in converging to the appropriate lower and upper level fronts can be increased. Only one Pareto-optimal point from each participating lower level problem qualifies to be on the upper level front. For our study here, we choose $K = 5$ and $L = 4$ (an overall 10-variable problem).

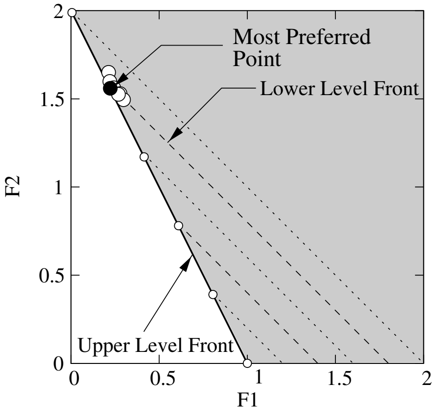


Fig. 5. Pareto-optimal front for problem DS4. Final parent population members have been shown close to the most preferred point.

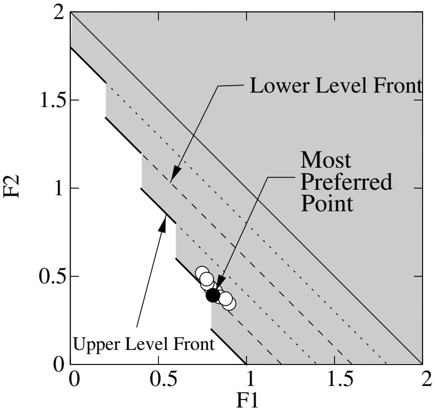


Fig. 6. Pareto-optimal front for problem DS5. Final parent population members have been shown close to the most preferred point.

Table 4. Total function evaluations for the upper and lower level (21 runs) for DS4

Algo. 1, Algo. 2, Savings	Best		Median		Worst	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE	Total LL FE	Total UL FE
HBLEMO	1,356,598	38,127	1,435,344	53,548	1,675,422	59,047
PI-HBLEMO	149,214	5,038	161,463	8,123	199,880	8,712
$\frac{HBLEMO}{PI-HBLEMO}$	9.09	7.57	8.89	6.59	8.38	6.78

The Pareto-front, most-preferred point and the final population members from a particular run are shown in Figure 5. Table 4 presents the function evaluations required by PI-HBLEMO to produce the final solution and the function evaluations required by HBELMO to produce an approximate Pareto-front.

6.5 Problem DS5

Problem DS5 has been taken from (9). A point on the Pareto-optimal front of the test problem is chosen as the most-preferred point and then the PI-HBLEMO algorithm is executed to obtain a solution close to the most preferred point. This problem is similar to problem DS4 except that the upper level Pareto-optimal front is constructed from multiple points from a few lower level Pareto-optimal fronts. There are $K + L + 1$ real-valued variables in this problem as well:

$$\begin{array}{ll}
\text{Minimize} \quad \mathbf{F}(\mathbf{x}, \mathbf{y}) = & \text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \mathbf{f}(\mathbf{x}) = \\
\begin{pmatrix} (1 - x_1)(1 + \sum_{j=2}^{K} x_j^2) y_1 \\ x_1(1 + \sum_{j=2}^{K} x_j^2) y_1 \end{pmatrix}, & \left\{ \begin{pmatrix} (1 - x_1)(1 + \sum_{j=K+1}^{K+L} x_j^2) y_1 \\ x_1(1 + \sum_{j=K+1}^{K+L} x_j^2) y_1 \end{pmatrix} \right\}, \\
G_1(\mathbf{x}) = (1 - x_1) y_1 + \frac{1}{2} x_1 y_1 - 2 + \frac{1}{5} \left[5(1 - x_1) y_1 + 0.2 \right] \geq 0, & \\
[\cdot] \text{ denotes greatest integer function,} & \\
-1 \leq x_1 \leq 1, \quad 1 \leq y_1 \leq 2, & \\
-(K + L) \leq x_i \leq (K + L), \ i = 2, \ldots, (K + L). &
\end{array} \tag{9}$$

For the upper level Pareto-optimal front, $x_i = 0$ for $i = 2, \ldots, (K + L)$, $x_1 \in [2(1 - 1/y_1), 2(1 - 0.9/y_1)]$, $y_1 \in \{1, 1.2, 1.4, 1.6, 1.8\}$ (Figure 6). For this test problem we have chosen $K = 5$ and $L = 4$ (an overall 10-variable problem). This problem has similar difficulties as in DS4, except that only a finite number of y_1 qualifies at the upper level Pareto-optimal front and that a consecutive set of lower level Pareto-optimal solutions now qualify to be on the upper level Pareto-optimal front.

The Pareto-front, most-preferred point and the final population members from a particular run are shown in Figure 6. Table 5 presents the function evaluations required by PI-HBLEMO to produce the final solution and the function evaluations required by HBELMO to produce an approximate Pareto-front.

Table 5. Total function evaluations for the upper and lower level (21 runs) for DS5

Algo. 1, Algo. 2, Savings	Best		Median		Worst	
	Total LL FE	Total UL FE	Total LL FE	Total UL FE	Total LL FE	Total UL FE
HBLEMO	1,666,953	47,127	1,791,511	56,725	2,197,470	71,246
PI-HBLEMO	168,670	5,105	279,568	6,269	304,243	9,114
$\frac{HBLEMO}{PI-HBLEMO}$	9.88	9.23	6.41	9.05	7.22	7.82

7 Accuracy and DM Calls

Table 6 represents the accuracy achieved and the number of decision maker calls required while using the PI-HBLEMO procedure. In the above test problems the most preferred point which the algorithm is seeking is pre-decided and the value function emulating the decision maker is constructed. When the algorithm terminates it provides the best achieved point. The accuracy measure is the Euclidean distance between the best point achieved and the most preferred point. It can be observed from the results of the PI-HBLEMO procedure that preference information from the decision maker leads to a high accuracy (Table 6) as well as huge savings (Table 1,2,3,4,5) in function evaluations. Producing the entire front using the HBLEMO procedure has its own merits but it comes with a cost of huge function evaluations and there can be instances when the entire set of close Pareto-optimal solutions will be difficult to achieve even after high number of evaluations. The accuracy achieved using the HBLEMO procedure has been reported in the brackets; the final choice made from a set of close Pareto-optimal solutions will lead to a poorer accuracy than a progressively interactive approach.

Table 6. Accuracy and the number of decision maker calls for the PI-HBLEMO runs (21 runs). The distance of the closest point to the most preferred point achieved from the HBLEMO algorithm has been provided in the brackets.

		Best	Median	Worst
DS1	Accuracy	0.0426 (0.1203)	0.0888 (0.2788)	0.1188 (0.4162)
	DM Calls	12	13	29
DS2	Accuracy	0.0281 (0.0729)	0.0804 (0.4289)	0.1405 (0.7997)
	DM Calls	12	15	25
DS3	Accuracy	0.0498 (0.0968)	0.0918 (0.3169)	0.1789 (0.6609)
	DM Calls	7	17	22
DS4	Accuracy	0.0282 (0.0621)	0.0968 (0.0981)	0.1992 (0.5667)
	DM Calls	7	15	23
DS5	Accuracy	0.0233 (0.1023)	0.0994 (0.1877)	0.1946 (0.8946)
	DM Calls	7	14	22

8 Conclusions

There are not many approaches yet to handle multi-objective bilevel problems. The complexity involved in solving a bilevel multi-objective problem has deterred researchers, keeping the area unexplored. The Hybrid Bilevel Evolutionary Multi-objective Optimization Algorithm is one of the successful procedures towards handling a general bilevel multi-objective problem. However, the procedure involves heavy computation, particularly at the lower level, to produce the entire Pareto-optimal set of solutions at the upper level.

In this paper, the Hybrid Bilevel Evolutionary Multi-objective Optimization Algorithm has been blended with a progressively interactive technique. An evaluation of the Progressively Interactive HBLEMO (PI-HBLEMO) technique against the HBLEMO procedure shows an improvement in terms of function evaluations as well as accuracy. The savings in function evaluations at the lower as well as upper level is in the range of six to ten times. This is a significant improvement particularly for cases where a function evaluation is computationally very expensive. Moreover, for problems where EMOs tend to suffer in converging towards the front, a progressively interactive approach provides a viable solution to such problems and leads to a higher accuracy. Therefore, an integration of the progressively interactive procedure with an EMO algorithm offers a dual advantage of reduced function evaluations and increased accuracy. Such kind of an integrated procedure, with EMO algorithm's parallel search and concepts from the field of MCDM, can be a useful tool to efficiently handle difficult multi-objective problems. The power of such an amalgamation has been shown in this study with its successful application on the challenging domain of multi-objective bilevel problems.

Acknowledgements

The author would like to thank Prof. Kalyanmoy Deb, Prof. Pekka Korhonen and Prof. Jyrki Wallenius for their useful feedback and discussions which led to this work.

References

[1] Byrd, R.H., Nocedal, J., Waltz, R.A.: KNITRO: An integrated package for nonlinear optimization, pp. 35–59. Springer, Heidelberg (2006)

[2] Calamai, P.H., Vicente, L.N.: Generating quadratic bilevel programming test problems. ACM Trans. Math. Software 20(1), 103–119 (1994)

[3] Colson, B., Marcotte, P., Savard, G.: An overview of bilevel optimization. Annals of Operational Research 153, 235–256 (2007)

[4] Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)

[5] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)

[6] Deb, K., Sinha, A.: Constructing test problems for bilevel evolutionary multi-objective optimization. In: 2009 IEEE Congress on Evolutionary Computation (CEC-2009), pp. 1153–1160. IEEE Press, Los Alamitos (2009)

[7] Deb, K., Sinha, A.: An evolutionary approach for bilevel multi-objective problems. In: Cutting-Edge Research Topics on Multiple Criteria Decision Making, Communications in Computer and Information Science, vol. 35, pp. 17–24. Springer, Berlin (2009)

[8] Deb, K., Sinha, A.: Solving bilevel multi-objective optimization problems using evolutionary algorithms. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 110–124. Springer, Heidelberg (2009)

[9] Deb, K., Sinha, A.: An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. Evolutionary Computation Journal 18(3), 403–449 (2010)

[10] Deb, K., Sinha, A., Korhonen, P., Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. IEEE Transactions on Evolutionary Computation 14(5), 723–739 (2010)

[11] Dempe, S., Dutta, J., Lohse, S.: Optimality conditions for bilevel programming problems. Optimization 55(5-6), 505–524 (2006)

[12] Eichfelder, G.: Solving nonlinear multiobjective bilevel optimization problems with coupled upper level constraints. Technical Report Preprint No. 320, Preprint-Series of the Institute of Applied Mathematics, Univ. Erlangen-Närnberg, Germany (2007)

[13] Fudenberg, D., Tirole, J.: Game theory. MIT Press, Cambridge (1993)

[14] Geoffrion, A.M.: Proper efficiency and theory of vector maximization. Journal of Mathematical Analysis and Applications 22(3), 618–630 (1968)

[15] Halter, W., Mostaghim, S.: Bilevel optimization of multi-component chemical systems using particle swarm optimization. In: Proceedings of World Congress on Computational Intelligence (WCCI 2006), pp. 1240–1247 (2006)

[16] Koh, A.: Solving transportation bi-level programs with differential evolution. In: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 2243–2250. IEEE Press, Los Alamitos (2007)

[17] Mathieu, R., Pittard, L., Anandalingam, G.: Genetic algorithm based approach to bi-level linear programming. Operations Research 28(1), 1–21 (1994)

[18] Oduguwa, V., Roy, R.: Bi-level optimization using genetic algorithm. In: Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS 2002), pp. 322–327 (2002)

[19] Sinha, A., Deb, K.: Towards understanding evolutionary bilevel multi-objective optimization algorithm. In: IFAC Workshop on Control Applications of Optimization (IFAC 2009), vol. 7. Elsevier, Amsterdam (2009)

[20] Sinha, A., Deb, K., Korhonen, P., Wallenius, J.: Progressively interactive evolutionary multi-objective optimization method using generalized polynomial value functions. In: 2010 IEEE Congress on Evolutionary Computation (CEC 2010), pp. 1–8. IEEE Press, Los Alamitos (2010)

[21] Vicente, L.N., Calamai, P.H.: Bilevel and multilevel programming: A bibliography review. Journal of Global Optimization 5(3), 291–306 (2004)

[22] Wang, F.J., Periaux, J.: Multi-point optimization using gas and Nash/Stackelberg games for high lift multi-airfoil design in aerodynamics. In: Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001), pp. 552–559 (2001)

[23] Wang, Y., Jiao, Y.-C., Li, H.: An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 35(2), 221–232 (2005)

[24] Yin, Y.: Genetic algorithm based approach for bilevel programming models. Journal of Transportation Engineering 126(2), 115–120 (2000)

[25] Zhang, G., Liu, J., Dillon, T.: Decentralized multi-objective bilevel decision making with fuzzy demands. Knowledge-Based Systems 20, 495–507 (2007)

[26] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K.C., Tsahalis, D.T., Périaux, J., Papailiou, K.D., Fogarty, T. (eds.) Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100. International Center for Numerical Methods in Engineering (Cmine) (2001)

Multi-objective Phylogenetic Algorithm: Solving Multi-objective Decomposable Deceptive Problems

Jean Paulo Martins, Antonio Helson Mineiro Soares,
Danilo Vasconcellos Vargas, and Alexandre Cláudio Botazzo Delbem

Institute of Mathematics and Computer Science,
University of São Paulo, São Carlos, SP, Brazil
{jean,ahms,vargas,acbd}@icmc.usp.br

Abstract. In general, Multi-objective Evolutionary Algorithms do not guarantee find solutions in the *Pareto-optimal set*. We propose a new approach for solving decomposable deceptive multi-objective problems that can find all solutions of the *Pareto-optimal set*. Basically, the proposed approach starts by decomposing the problem into subproblems and, then, combining the found solutions. The resultant approach is a Multi-objective Estimation of Distribution Algorithm for solving relatively complex multi-objective decomposable problems, using a probabilistic model based on a phylogenetic tree. The results show that, for the tested problem, the algorithm can efficiently find all the solutions of the *Pareto-optimal set*, with better scaling than the hierarchical Bayesian Optimization Algorithm and other algorithms of the state of art.

1 Introduction

Techniques of search and optimization based in the Theory of Evolution as the Evolutionary Algorithms (EA) are distinguished in the solution of complex problems of optimization that possess some objective functions [3,7]. In such problems, in general, the objectives are conflicting, and there is not only one optimal solution that satisfies equally every objective, thus a set of solutions should be chose to attend the objectives of the problem.

Another important and relatively recent research area on EAs is focused on the developing of Estimation of Distribution Algorithms (EDAs) [17,18]. The main idea of the EDAs is the creation of a model that represents the relations between variables in a population (Building Blocks – BBs), through this model the disruption of BBs can be avoided, allowing the generation of better solutions and fast convergence.

In literature some EDAs are found to solve multi-objective problems as the meCGA [22] and the mohBOA [19], but these has in the construction of their models an impediment, resulting in a great number of functions evaluations. On the other hand, faster EDAs in general construct poorer models limiting their performance for large-scale multimodal problems. In fact, there is a trade-off between model quality and running time to construct the model itself.

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 285–297, 2011.

Fortunately, several methods to reconstruct phylogenies (models describing the relationship among taxa as, for example, the species features) were developed in the last century. The phylogeny literature [9,21] shows that one of these methods, the Neighbor Joining (NJ) [20,23], is an adequate trade-off between quality and efficiency. This paper proposes a Multi-objective Estimation of Distribution Algorithm (MOEDA) based on phylogenetic models, using the NJ as a method to guarantee solutions in the *Pareto-optimal front* with reduced number of functions evaluations. This MOEDA is the multi-objective variation of the ΦGA (Phylo-Genetic Algorithm) [15,25,26] called moΦGA.

The remaining of the paper is organized as follows. Section 2 reviews fundamental concepts on Multi-objective Evolutionary Algorithms (MOEAs). Section 3 introduces the MOEDAs. Section 4 describes the Multi-objective Phylogenetic Algorithm (moΦGA). Section 5 shows tests and results within moΦGA and Sect. 7 concludes the paper.

2 Multi-objective Evolutionary Algorithms

Multi-objective Evolutionary Algorithms is a well established field within Evolutionary Computation that deal with problems with multiple objectives (MOP - Multi-objective Optimization Problem). In such problems, the solutions are defined in relation to a set of objectives, so that each objective contributes to the solution's quality of the MOP.

Conflicting objectives are common in a MOP. In these cases, an increase for a particular objective is often limited by the decline in value of another objective. As such, in accordance with [5] there is basically two ways for solving multi-objective problems :

1. Preference-based methods;
2. Generating methods.

The *Preference-based methods* are normally formalized as the application of weights to the objective in some way, yielding a single-objective function. Each weight indicates the importance of the objective for the whole problem, thus, we have a composite function as (1), where x represents a solution to the whole problem and α_i is the weight of every single-objective function $f_i(x)$.

$$g(x) = \sum_{i=1}^{M} \alpha_i f_i(x). \tag{1}$$

In that way, the multi-objective problem can be solved as a mono-objective problem. However, this approach requires knowledge of the weights for each objective, which is often not available or known. Moreover, those type of methods are not designed to find a family of solutions.

The *Generating methods* use a population of individuals to evolve multiple objectives at the same time. Using various strategies such as applying a changing selection criterion, choosing individuals based on their pareto optimality, among others. These type of algorithms commonly aim at finding the *Pareto-optimal set*, since it allows the obtention of multiple optimal solutions concurrently without the need to balance goals.

The *Pareto-optimal front* from a given problem is defined as a set of all *Pareto-optimal* solutions. A solution is considered *Pareto-optimal* according to the concept of *dominance* [5], where a solution x_k dominates a solution x_l if:

1. The solution x_k is no worse (say the operator $\prec$ denotes worse and $\succ$ denotes better) than x_l in all objectives, or $f_i(x_k) \nprec f_i(x_l)$, for $i = 1, \ldots, k$ objectives,
2. The solution x_k is strictly better than x_l in at least one objective, or $f_{\bar{i}}(x_k) \succ f_{\bar{i}}(x_l)$, for at least one $\bar{i} \in \{1, \ldots, k\}$.

So, the Pareto-optimal set consists of non-dominated solutions from the solution space, while the Pareto-optimal front is the edge, at the objective space, composed by all non-dominated solutions.

Several MOEAs have been proposed based on the concept of dominance and applied to real problems (for example the Non-dominated Sorting (NSGA) [6] and the Strength Pareto EA (SPEA) [27]). However, some studies show problems regarding the scalability and time-convergence of these algorithms [5].

In Sect. 3 an alternative for the use of MOEAs is described, although there are also limitations on the scalability of this new approach [22][11], it presents significant advantages as the proximity of *Pareto-optimal front* and the number of evaluations.

3 Multi-objective Estimation of Distribution Algorithms

The Multi-objective Estimation of Distribution Algorithms differ from other MOEAs for not limiting themselves to raw information from a population. They use the selected individuals from a population as samples of an unknown probabilistic distribution and generate probabilistic models from these samples. The models should approximate the probabilistic distribution of the values of the variables or sub-sets of variables (strong correlation among variables, Building Blocks – BBs). In this way, a model is also a representation of the selected population itself (the used samples) and of other possible solutions that are related to the model but were not among the selected individuals [17,18,22].

The common reproduction operators are not used by MOEDAs since new individuals can be generated by directly sampling from the model. Moreover, if such model can identify the correct BBs of a problem, the combination of improbable values of variables can be avoided. An MOEDA that adequately estimates BBs can solve relatively large-scale complex problems.

However, as more representative is the model (more BBs it correctly estimates), more computationally complex is the algorithm to construct it. Thus, there is a trade-off between the efficiency of the algorithm for construction of a model and the accuracy of the model [17]. As a consequence, MOEDAs are directly affected by the algorithm efficiency and the accuracy of the model they generate. Thus, an MOEDA with an adequate commitment between efficiency and accuracy would be a relevant contribution. Furthermore, several classes of important problems involving multimodal objective functions, large-scale instances, and solutions in real-time could be adequately solved using such MOEDA.

This paper proposes an MOEDA based on a method that has not been used in MOEDAs, the phylogenetic reconstruction [9]. This method enables an adjustable trade-off between the accuracy of the generated model (Phylogenetic Tree) and the computational efficiency to construct it.

4 Multi-objective Phylogenetic Algorithm

The Multi-objective Phylo-Genetic Algorithm (moΦGA) is an MOEDA based on the ΦGA, extending its features to find the *Pareto-optimal* set, i.e., solving multi-objective problems. It uses a phylogenetic tree reconstruction (NJ) method to generate a phylogenetic tree for each problem separately. And from the analysis of each tree through the uncertainty variation [15,26], the moΦGA can exactly determine the BBs of a separable deceptive multi-objective problem. This algorithm is synthesized in the diagram from Fig.1, where every branch is associated to one objective (Obj_1, Obj_2, $\ldots$, Obj_M).

The moΦGA starts with a population of size P with random individuals and applies Θ tournament selections defined by (2).

$$\Theta = \Phi P \tag{2}$$

Where Φ is a variable already used in previous ΦGAs. And it can adjust a trade-off between number of avaliations and the running time of the algorithm [15,25,26].

For a selected population (of a respective objective), a method calculates a distance matrix (Sect. 4.2), required by the NJ. The moΦGA uses Mutual Information (Sect. 4.1) to calculate the distance matrix. Then, NJ is applied to create a phylogenetic tree for each objective separately. And the same procedure is applied to all objectives.

Then the BBs are identified based on an uncertainty criteria of the NJ method. That is, considering the difference between the average and minimum distance in the matrix M (see Section 4.2 for details of the matrix M), if we call the average minus the minimum distance in the M in a specific t iteration α_t, the relative rate it varies can be approximate by the following function fraction:

$$\frac{\alpha_t - \alpha_{t-1}}{\alpha_t}. \tag{3}$$

Disconsidering the points with α_t near 0, the highest point $t*$ of this fraction represent the stopping criteria for the NJ. Therefore, every internal node created by the NJ after the point $t*$ should be discarded and the remaining subtrees will form the BBs [26],[15].

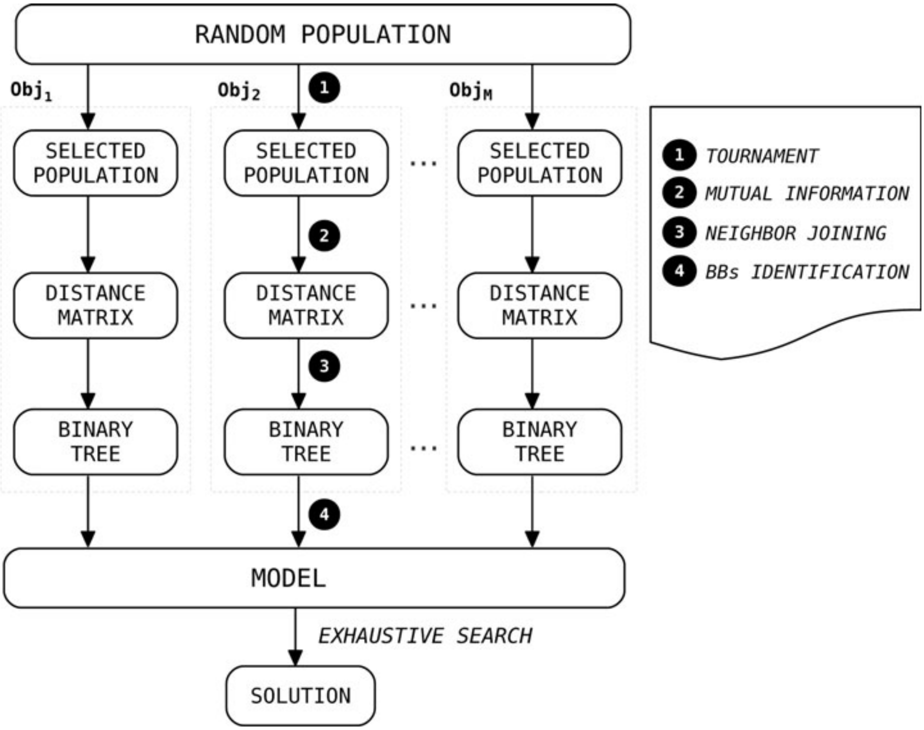


Fig. 1. Basic steps of the moΦGA for separable deceptive problems

Sequentially, each objective is combined as clusters (excluding equal clusters of variables), leading to a small set of BBs. Finally, an exhaustive search is applied at each BB from the set of BBs to find the *Pareto-optimal* for each BB. The combination of the *Pareto-optimal* solutions at each obtained BB in the tests was able to compose not just some or the solutions with different values of fitness, but all the possible combinations of solutions in the *Pareto-optimal front*.

This type of result was possible, because moΦGA focuses on solving the linkage learning (the interdependence between the variables). This contrasts with the results of other usual multi-objective Population-based Algorithms, which are able to find most but not all solutions. This may also shed light to a different focus on tackling these types of problems, which is to focus on decomposing them and not on searching solely for the solutions. The solution of the decomposition, as showed by this article, may make the searching process easier.

The complete description of the algorithm are explained in the sections as follows, Sect. 4.1 describe how the distance matrix is built, necessary for the NJ (reconstruction of phylogenetic trees method). Which is presented in Sect. 4.2.

4.1 Distance Metric

In order to construct the distance matrix D, a metric capable of extracting the correlation between variables is desirable. Mutual Information [14] is a measure of similarity between two variables which has been successfully applied to the development of EDAs. Equation (4) describe it, where X and Y are two random variables, $p_{xy}(X, Y)$ is the joint probability of $(X = x, Y = y)$; and $p_x(X)$ and $p_y(Y)$ are the marginal probability of $X = x$ and $Y = y$, respectivelly.

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p_{xy}(X, Y) \ log \frac{p_{xy}(X, Y)}{p_x(X) p_y(Y)} \,. \tag{4}$$

However, the Mutual Information is not a metric, since it does no satisfy strictly the triangle inequality. Thus (5), as described in [14] has been widely used in the literature [1,2,8,13] specially because it satisfies the triangle inequality, non-negativity, indiscernibility and symmetric properties. In (5), $H(X, Y)$ is the entropy of the pair (X, Y).

$$D(X, Y) = H(X, Y) - I(X, Y) \,. \tag{5}$$

The metric (5) is also an universal metric, in the sense that if any non-trivial distance measure places X and Y close, then it will also be judged close according to $D(X, Y)$ [14].

Once the phylogenetic tree is constructed, the subtrees that represents the correlated variables (these subtrees are called clados in Biology), and consequently the Building Blocks, need to be identified, and this identification is an empirical procedure that depends on *adhoc* knowledge about the problem. This problem was solved for the use of the NJ by the ΦGA and explained at [26,15,25].

4.2 Neighbor Joining

In this context, the Neighbor Joining can be viewed as a hierarchical clustering method [4,12,16] that uses a distance matrix D to create a phylogenetic tree [20,23]. The distance matrix is used as information about the relationship between variables and the tree is built in such way that more related variables must be closer in the tree.

The algorithm starts with all variables connected to an unique internal node, labeled “0”, composing a star tree of $n + 1$ nodes, where n is the number of variables, Fig. 2 illustrates a star tree.

For every leaf node $i \in N = \{1, \ldots, n\}$, the net divergence, R_i, is calculated by (6), which is the sum of all distances referent to node i. From this, a new matrix M is calculated using D and net divergences (7), where D_{ij} is the element at position (i, j) in D.

$$R_i = \sum_{j \in N, j \neq i} D_{ij} \,. \tag{6}$$

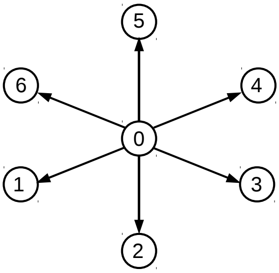


Fig. 2. Star tree of a problem with 6 variables

$$M_{ij} = D_{ij} - \frac{R_i + R_j}{n-2}\,. \tag{7}$$

M is very important for the NJ success since the joining nodes will derive from it. The two nodes, i and j, corresponding to the smallest M_{ij}, are removed from the star tree. Then, we insert a new node u, connected to i and j, beginning the formation of a binary tree.

To calculate the branch length from u to i and j, (8) and (9) are used, where S_{ij} is the branch length between i and j.

$$S_{iu} = \frac{D_{ij}}{2} + \frac{R_i - R_j}{2(n-2)}, \tag{8}$$

$$S_{ju} = \frac{D_{ij}}{2} + \frac{R_j - R_i}{2(n-2)}\,. \tag{9}$$

Afterward, it is calculated the distance from the other variables to u in order to fulfill a new D (without columns related to i and j and with a new column for u). Such calculi is defined by (10), where k is necessarily a node different from u in D.

$$D_{ku} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}, \forall k \neq u. \tag{10}$$

D has decreased its size from $n \times n$ to $(n-1) \times (n-1)$. Then, this new D is used to calculate a new M. The whole process is repeated until the size of D reaches 2. Then, the two remaining nodes are joined together creating the last internal node, which is set as the root of the tree. Lastly, the node "0" (that primarily connected every node in a star tree) is removed from the tree.

5 Test Problems and Results

The moΦGA was tested with a relatively complex multi-objective function from the literature [22], considering all the test factors: problem size, population size and number of executions. This multi-objective function is a combination of two objectives and is defined by (11) and (12), both being linearly separable fully deceptive problems with conflicting objectives, where x represents a solution to the problem, m is the number of BBs and $k = 5$ is the trap size.

The *trap*5 and the *inv-trap*5 are difficult problems, because standard crossing over operators can not solve it, unless the bits of each partition are placed close to each other. Mutation operators are also very inefficient for solving them, requiring $O(n^5 \log n)$ evaluations [24].

$$\max\ f_{trap5}(x),\ x \in \{0,1\}^{mk}, \tag{11}$$

$$f_{trap5}(x) = \sum_{i=0}^{m-1} trap5(x_{ki} + x_{ki+1} + ... + x_{ki+k-1}),$$

$$trap5(u) = \begin{cases} 5 & \text{if } u = 5, \\ 4 - u & \text{if } u < 5\,. \end{cases}$$

$$\max\ f_{inv-trap5}(x),\ x \in \{0,1\}^{mk}, \tag{12}$$

$$f_{inv-trap5}(x) = \sum_{i=0}^{m-1} inv\text{-}trap5(x_{ki} + x_{ki+1} + ... + x_{ki+k-1}),$$

$$inv\text{-}trap5(u) = \begin{cases} 5 & \text{if } u = 0, \\ u - 1 & \text{if } u > 0\,. \end{cases}$$

Considering a binary representation for the solutions, (11) has an optimum global solution consisting uniquely of ones, whereas (12) has an optimum global solution consisting uniquely of zeros. The Fig. 3 represents the relation between them in a string of 5 bits.

The moΦGA was used to find solutions in the *Pareto-optimal front*. To confirm the solutions quality, the entire solution space was identified, enabling the confirmation that the solutions found lies in the *Pareto-optimal front*.

Some parameters used for the experiments are described in Table 1, also are used a tournament of size 16, and 30 runs for each experiment.

Figures 4 and Fig. 5 presents the space of solutions and the *Pareto-optimal set* found respectively for problems of size 30 and 50.

Those tests evidence the robustness of the algorithm, because it was able to find the entire *Pareto-optimal set* for all the tests presented. Even with the

Table 1. Population sizes obtained by the Bisection Method [10]

Problem size	30	50	100
Population size	3843	7704	22543

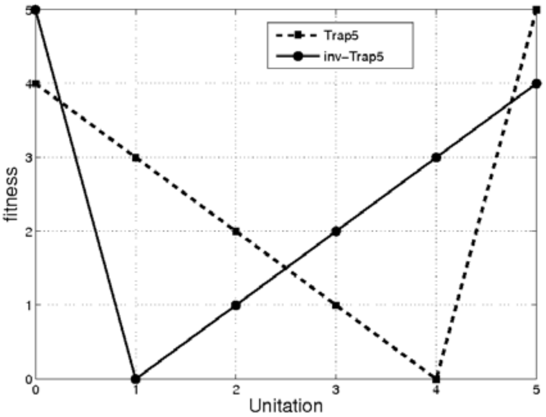


Fig. 3. Functions $trap_5$ and $inv - trap_5$

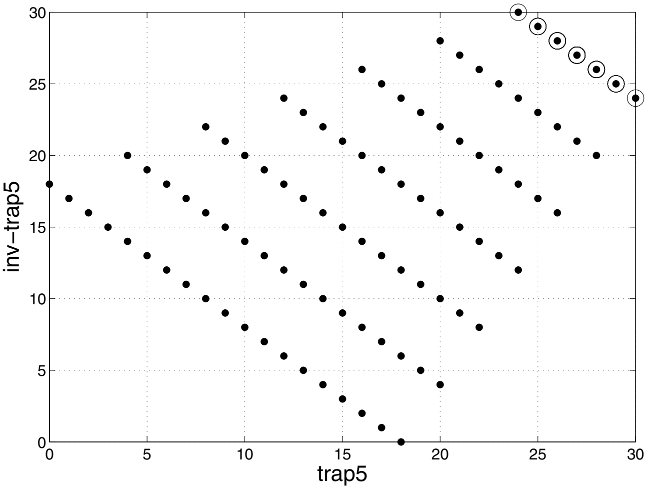


Fig. 4. Problem of size 30. The circled points are both the solutions found and the complete *Pareto-optimal front.*

amount of possible solutions growing exponentially with the size of the problem [22].

The *trap*5 vs. *inv-trap*5 problem was also tested by [19] and ours experiments were done using the same parameters. The results obtained by them and the results found by us (applying moΦGA to problems of size 30, 50 and 100) are represented in the Fig. 6.

By observing Fig. 6 it is possible to perceive that the number of evaluations used by moΦGA has a lower slope than the algorithms described in [19], where

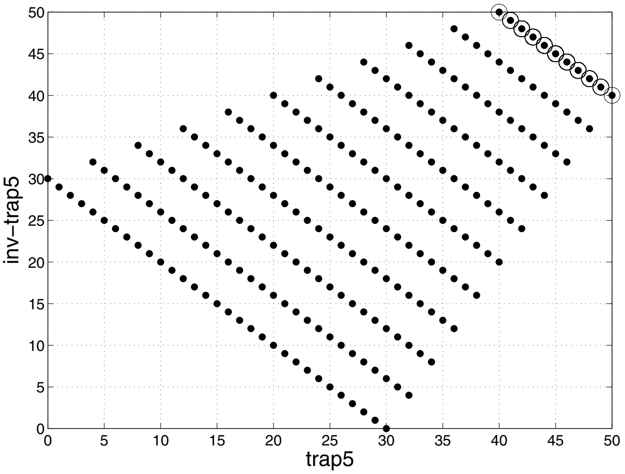


Fig. 5. Problem of size 50. The circled points are both the solutions found and the complete *Pareto-optimal front.*

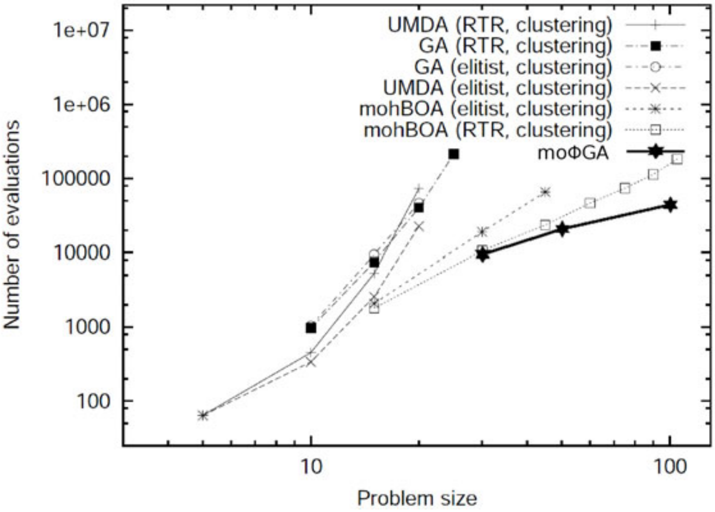


Fig. 6. A comparison of *trap*5 vs. *inv-trap*5 problem. This image was first published in [19] and edited to include the mo*Φ*GA results.

multi-objective hBOA had better results than the others algorithms tested, this can be an indication of the minor complexity of the moΦGA in relation to the other algorithms from the literature.

6 Future Work

This article sheds light on an approach of how to solve efficiently relative complex decomposable deceptive problems. The trap-invtrap problem is an important one, because it points out deficiencies of many discrete multi-objective EAs, however, it is not enough, lots of another experiments need to be done before the approach can be completely validated.

Thus, there are important points to be extended in future versions of moΦGA to enable the algorithm to solve many-objective problems, multi-objective problems with different Building Block sizes, problems in continuous search spaces, hierarchical problems and so on.

7 Conclusions

A great variety of problems exists in the real world, which naturally demand efficient solutions of multi-objective problems. For such problems the MOEDAs were proven to achieve the most prominent solutions. This paper proposes a MOEDA, the moΦGA, which can adjust the trade-off between running time and functions evaluations, by the variation of parameter Φ.

Moreover, in all the problems tested, the moΦGA surpassed the mohBOA in terms of less function evaluations. Specially when the problems in question increased in size. In fact, the moΦGA finds the entire set of solutions that compose the *Pareto-optimal front*, not only some solutions as most of the algorithms from the literature. This is made possible by approaching problems in two clear stages:

1. Decomposition Stage
2. Optimization Stage

The importance of the approach is demonstrated specially well in the proposed algorithm, because even with a simple optimization stage, the moΦGA achieved state of art results. The same approach was validated in other difficult problems by other ΦGA algorithms [15,26].

The promising results presented by the moΦGA motivates new variants of it, as for example, an algorithm with a more robust optimization stage, as well as, an extension to deal with continuous global optimization problems and hierarchical deceptive problems.

References

1. Aghagolzadeh, M., Soltanian-Zadeh, H., Araabi, B., Aghagolzadeh, A.: A hierarchical clustering based on mutual information maximization. In: IEEE International Conference on Image Processing, ICIP 2007, vol. 1 (2007)

2. Aporntewan, C., Ballard, D., Lee, J.Y., Lee, J.S., Wu, Z., Zhao, H.: Gene hunting of the Genetic Analysis Workshop 16 rheumatoid arthritis data using rough set theory. In: BMC Proceedings, vol. 3, p. S126. BioMed Central Ltd (2009)
3. Coello, C.A.C., Zacatenco, S.P., Pulido, G.T.: Multiobjective optimization using a micro-genetic algorithm (2001)
4. Day, W., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of classification 1(1), 7–24 (1984)
5. Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary computation 7(3), 205–230 (1999)
6. 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)
7. Deb, K.: Multi-objective optimization using evolutionary algorithms (2001)
8. Dionísio, A., Menezes, R., Mendes, D.A.: Entropy-based independence test. Nonlinear Dynamics 44(1), 351–357 (2006)
9. Felsenstein, J.: Inferring Phylogenies, vol. 266. Sinauer Associates (2003)
10. Harik, G.: Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. Ph.D. thesis, The University of Michigan (1997)
11. Hughes, E.: Evolutionary many-objective optimisation: many once or one many? In: The 2005 IEEE Congress on Evolutionary Computation, vol. 1, pp. 222–227. IEEE, Los Alamitos (2005)
12. Johnson, S.: Hierarchical clustering schemes. Psychometrika 32(3), 241–254 (1967)
13. Kraskov, A.: Synchronization and Interdependence Measures and Their Application to the Electroencephalogram of Epilepsy Patients and Clustering of Data. Report Nr. NIC series 24 (2008)
14. Kraskov, A., Stogbauer, H., Andrzejak, R., Grassberger, P.: Hierarchical clustering based on mutual information. Arxiv preprint q-bio/0311039 (2003)
15. de Melo, V.V., Vargas, D.V., Delbem, A.C.B.: Uso de otimização contínua na resolução de problemas binários: um estudo com evolução diferencial e algoritmo filo-genético em problemas deceptivos aditivos.. In: 2^a Escola Luso-Brasileira de Computação Evolutiva (ELBCE), APDIO (2010)
16. Morzy, T., Wojciechowski, M., Zakrzewicz, M.: Pattern-oriented hierarchical clustering. In: Eder, J., Rozman, I., Welzer, T. (eds.) ADBIS 1999. LNCS, vol. 1691, pp. 179–190. Springer, Heidelberg (1999)
17. Pelikan, M., Goldberg, D., Lobo, F.: A survey of optimization by building and using probabilistic models. Computational optimization and applications 21(1), 5–20 (2002)
18. Pelikan, M., Sastry, K., Cantu-Paz, E.: Scalable optimization via probabilistic modeling: From algorithms to applications. Springer, Heidelberg (2006)
19. Pelikan, M., Sastry, K., Goldberg, D.: Multiobjective hBOA, clustering, and scalability. In: Genetic And Evolutionary Computation Conference: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 663–670. Association for Computing Machinery, Inc., New York (2005)
20. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4(4), 406 (1987)
21. Salemi, M., Vandamme, A.M.: The Phylogenetic Handbook: A Practical Approach to DNA and Protein Phylogeny, vol. 16. Cambridge University Press, Cambridge (2003), http://doi.wiley.com/10.1002/ajhb.20017
22. Sastry, K., Goldberg, D., Pelikan, M.: Limits of scalability of multiobjective estimation of distribution algorithms. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 3, pp. 2217–2224. IEEE, Los Alamitos (2005)

23. Studier, J., Keppler, K.: A note on the neighbor-joining algorithm of Saitou and Nei. Molecular Biology and Evolution 5(6), 729 (1988)
24. Thierens, D.: Analysis and design of genetic algorithms. Katholieke Universiteit Leuven, Leuven (1995)
25. Vargas, D.V., Delbem, A.C.B.: Algoritmo filogenético. Tech. rep., Universidade de São Paulo (2009)
26. Vargas, D.V., Delbem, A.C.B., de Melo, V.V.: Algoritmo filo-genético. In: 2^a Escola Luso-Brasileira de Computação Evolutiva (ELBCE), APDIO (2010)
27. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (2002)

Multi-objective Optimization with Joint Probabilistic Modeling of Objectives and Variables

Hossein Karshenas, Roberto Santana, Concha Bielza, and Pedro Larrañaga

Computational Intelligence Group, School of Computer Science,
Technical University of Madrid, Campus de Montegancedo,
28660 Boadilla del Monte, Madrid, Spain
{hkarshenas,mcbielza,pedro.larranaga}@fi.upm.es,
roberto.santana@upm.es

Abstract. The objective values information can be incorporated into the evolutionary algorithms based on probabilistic modeling in order to capture the relationships between objectives and variables. This paper investigates the effects of joining the objective and variable information on the performance of an estimation of distribution algorithm for multi-objective optimization. A joint Gaussian Bayesian network of objectives and variables is learnt and then sampled using the information about currently best obtained objective values as evidence. The experimental results obtained on a set of multi-objective functions and in comparison to two other competitive algorithms are presented and discussed.

Keywords: Multi-objective Optimization, Estimation of Distribution Algorithms, Joint Probabilistic Modeling.

1 Introduction

Real-world problems usually include several criteria that should be fulfilled at the same time when trying to solve them. In many of these problems none of the criteria can be preferred over the others by the decision maker (the person who the optimization results are meant for) in order to apply single-function optimization techniques to solve them. On the other hand the criteria may be conflicting, i.e. trying to improve one of them will result in worse values for some other. Therefore it seems more reasonable to try solving them as Multi-objective Optimization Problems (MOPs).

Multi-Objective Evolutionary Algorithms (MOEAs) [1,6,29,34] have been successfully applied to many MOPs and obtained competitive results. Estimation of Distribution Algorithms (EDAs) [16,19,23,25] are proposed as a new computation paradigm based on evolutionary algorithms that replace the traditional recombination operators by learning and sampling a probabilistic model for advancing the search in solution space. Different Multi-objective EDAs (MEDAs) [20,26,32,33] have been proposed for solving MOPs. The main idea in these algorithms is to incorporate the selection and replacement strategies of MOEAs

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 298–312, 2011.

in the model-building framework of EDAs which will allow to utilize the power of EDAs for solving MOPs.

Although most of the study on MOPs has been focused on problems with a few number of objectives, very often practical optimization problems involve a large number of criteria. In addition, after the initial success of applying MOEAs to problems with two or a small number of objectives, efforts have been oriented towards investigating the scalability of these algorithms with respect to the number of objectives [8,13]. Therefore problems with many-objectives are receiving an increasing attention in the fields of decision making and multi-objective optimization. Another line of research related to many-objective problems is to reduce the optimization complexity by exploiting the relationships between the objectives. Objective reduction is one of the proposed methods that seeks the minimum objective subset by considering the conflicting and non-conflicting objectives [3,4,18].

In this paper an MEDA using a generalized Gaussian Bayesian Network (GBN) is proposed for solving MOPs. This probabilistic model not only allows capturing the dependencies between problem variables, but also permits modeling the dependencies between objective functions and the dependencies between objectives and variables which can be used for factorizing MOP into simpler subproblems. Furthermore it allows the insertion of information about good objective values to the model as evidence.

A similar idea has been used in the Evolutionary Bayesian Classifier-based Optimization Algorithm (EBCOA) [21,22] for single-objective optimization where a single class variable is introduced as a node in the Bayesian classifier models. However, there the class variable is inserted into the model with a fixed relation structure and only having a predefined limited number of different values used to classify the solutions to fitter and worse groups. The algorithm presented here extends the scope to multi-objective problems using a more general Bayesian network and includes the objective variables as the nodes in the model. The dependency structure between the objectives is also explored in the course of evolution, capturing the relation between the objectives of the problem.

The rest of this paper is organized as follows: in Section 2 the proposed EDA and its probabilistic model learning process are described in detail. The conducted experiments and their results are presented in Section 3. Section 4 gives the conclusive remarks and lines of future research.

2 Joint Modeling of Objectives and Variables

In a typical Evolutionary Algorithm (EA), the objective function values of solutions are used for selecting parents or replacing offspring, and beyond that the new-solution-generating part only relies on the information provided by variables. However if this new-solution-generator could exploit the objectives information, then it might be able to generate better solutions. In the case of EDAs, incorporating the objectives information to the probabilistic model can help finding out how solutions are involved in building fitter solutions and also to capture the relationship between objectives and the variables in the multi-objective context.

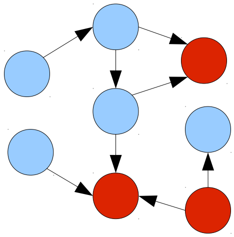


Fig. 1. An example of a joint model of 3 objectives and 5 variables

The deterministic functions given for a problem, link the objective values to variables. EDAs try to represent the problem structure by probabilistically approximating the dependencies between variables and then use them for obtaining new solutions with better objective values. Including the objectives in the model will allow the algorithm exploit the probabilistic approximation of the relationships learnt for them (e.g. based on the expected value of the objectives). This is especially useful in multi-objective problems with different, possibly conflicting, objective functions. Furthermore, the model structure makes it possible to identify redundancy, conditional independence, or other relationships between objectives. Fig. 1 shows a sample joint model that depicts different dependencies between variables and objectives.

The probabilistic model used in this paper is a Bayesian network. As it is usual for many continuous EDAs and because of its special analytical properties, it is assumed that the problem solutions follow a Gaussian distribution and therefore, the probabilistic model will be a Gaussian Bayesian Network (GBN). To learn the joint GBN, the strings of selected solutions are extended by appending the objective values of each solution to its variable values. Fig. 2 shows an overview of the algorithm. The main steps are each described next in detail.

2.1 Selection

The set of promising solutions are selected using a truncation scheme where given a factor τ and a population of N individuals, the selection operator first sorts the population and then selects the best τN solutions. One of the frequently used sorting algorithms in the multi-objective optimization is the non-dominated sorting introduced in [7]. The algorithm starts with sorting the solutions to different non-dominated fronts according to the dominance relation.

After grouping the population into several disjoint fronts, the solutions in each of the fronts are ranked using a crowding distance measure which favors those solutions that are in scattered areas of the front. The crowding distance computes how close each solution is to its neighbors with regard to different objective functions. To apply the truncation selection using non-dominated sorting, the non-dominated fronts are added to the set of selected solutions one by one according to their ranks and when a complete front can not be added anymore a subset of its solutions are selected according to their crowding distances. A slight modification to this algorithm is to select solutions one-by-one from the

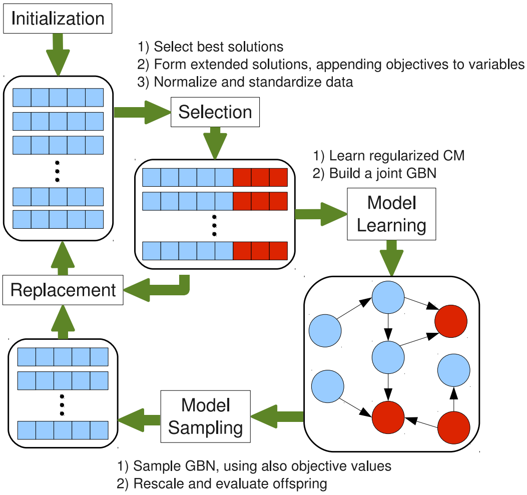

Initialization
1) Select best solutions
2) Form extended solutions, appending objectives to variables
3) Normalize and standardize data
Selection
1) Learn regularized CM
2) Build a joint GBN
Model Learning
Replacement
Model Sampling
1) Sample GBN, using also objective values
2) Rescale and evaluate offspring

Fig. 2. An overview of the proposed algorithm

last eligible front and after removing each selected individual, recomputing the crowding distance of all individuals still in that front. Although this modification will cause an increase in the computational cost, it can help to improve the diversity of the selected solutions [33].

Another sorting algorithm applied in this paper is based on Fitness Averaging (FA). It is a very simple and fast sorting mechanism for multi-objective optimization problems. Instead of treating different objective functions using dominance relation, a single fitness value is assigned to each solution which is the average of different objective values for that solution (after normalizing the objective values). Since this kind of fitness assignment gives a complete ordering between the solutions, selection can be performed using any desired selection scheme.

2.2 Model Learning

As a Bayesian network, the graphical structure in GBN is represented with a directed acyclic graph (DAG) and the conditional probability density of the continuous nodes are given by the following normal distribution [14,17]

$$p\left(x_i|pa(X_i)\right) = \mathcal{N}(\mu_i + \sum_{X_j \in Pa(X_i)} w_j(x_j - \mu_j), \sigma_i^2) \qquad (1)$$

where μ_i and σ_i are the parameters of the joint Gaussian distribution $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, $Pa(X_i)$ is the parent set of the ith variable according to the structure, and w_j are the weights of the conditional density function.

If the notation $(X_1, \ldots, X_p) = (V_1, \ldots, V_n, O_1, \ldots, O_m)$ is used to represent the joint extended vector of problem variables and objectives, then the following factorization shows the probability density encoded in the resulting GBN

$$p\left(v_1, \ldots, v_n, o_1, \ldots, o_m\right) = \prod_{i=1}^{n} p(v_i|pa(V_i)) \cdot \prod_{j=1}^{m} p(o_j|pa(O_j)) \qquad (2)$$

where $pa(V_i)$ and $pa(O_j)$ are the parents value setting of variable V_i and objective O_j respectively.

To further increase the learning accuracy of the probabilistic model, a regularization method is employed in estimating the parameters. The covariance matrix (CM) shrinkage technique discussed in [30] is applied to the covariance matrix computation of the joint Gaussian distribution before using it for learning the GBN. In this method the unbiased empirical estimation of the covariance matrix is combined with a diagonal matrix using a linear shrinkage. The shrinkage intensity is analytically computed using the statistical analysis of empirical covariance and correlation matrices. The regularized estimation increases the sparsity of the inverse covariance matrix which represents the relationships between variables.

Before passing the set of selected (extended) solutions to the model learning algorithm, they are standardized to have a mean of zero and a variance of one, in order to facilitate the parameter estimation [31]. The learning algorithm is based on the search and score strategy and employs a local search method for moving in the space of all possible DAG structures. It checks all edge additions, deletions and removals applicable to the current network structure and selects the one that results in the best improvement of the scoring metric and terminates if no further improvement is possible [5]. The scoring metric used here is a penalized log-likelihood metric with a Bayesian Information Criterion (BIC) penalization. The parameter stored for the node corresponding to the ith variable is the weight vector $\mathbf{w}$ needed for computing the conditional probability distribution of (1)

$$\mathbf{w} = \frac{1}{A} \frac{\Sigma_{(Pa(X_i),X_i)}}{A^T} \qquad (3)$$

where $\Sigma_{(K,L)}$ denotes the sub-matrix obtained from Σ by selecting the rows in set K and columns in set L, and the lower triangular matrix A is obtained from a *Cholesky* decomposition [27] of the parent part of the covariance matrix

$$AA^T = \Sigma_{(Pa(X_i),Pa(X_i))} \ . \qquad (4)$$

The condition for terminating the algorithm is to reach a maximum number of iterations. Therefore if the learning algorithm gets stuck in a local optimum point (or possibly the global optimum) before the number of iterations elapses, it will restart the search from another point randomly chosen in the DAGs space [31].

2.3 Model Sampling

The sampling procedure used in this study for the joint GBN is very similar to that of a typical Bayesian network sampling method except that special care is needed to handle objective nodes included in the model. The Probabilistic Logic Sampling (PLS) or Forward Sampling [14] works by first computing an ancestral or topological ordering between variables and then generating values for the variables according to this ordering, using the conditional probability distributions stored in the model.

Two strategies can be applied for sampling a joint probabilistic model: a) using the current best found values for the objectives, or b) generating dummy values for objectives in the process of sampling. A combination of these two methods is also possible. In the former case, the objective values of the best solutions of the population (i.e. the parents set) is passed to the sampling method to be used whenever an objective node has appeared as the parent of a variable node and its value is required for sampling. In the latter method, the objective nodes are completely treated as variable nodes and values are generated for them using the probabilities encoded in the model. Although these values are not used after sampling, and the new generated solutions need to be evaluated using objective functions, the dummy objective values can be used for generating new values for variable nodes and can lead to a more consistent sampling with regard to the learnt model. The results presented in this paper are obtained using the second method.

Since the new solutions are generated using a bell-shaped normal distribution, it may happen that some of the generated values fall outside of the domain of the variables. Therefore usually a repairment step becomes necessary after the new solutions are generated in the continuous EDAs based on Gaussian distribution. The repairing technique used in this paper is to reset those values that are out of the bound, to a random value in the acceptable domain of the variable. To increase the possibility of appropriate value reseting, the variables are both normalized and standardized before modeling takes place and the mean of each variable in each solution is computed according to the probabilistic model. Then the generated unacceptable value is replaced with a random value between the computed mean and the domain-bound (upper or lower) that was breached.

2.4 Replacement

The newly generated solutions should be incorporated into the original population for further exploitation in next generations. The *Elitist* replacement used in this paper selects the best solutions, in terms of Pareto dominance, of the current original and offspring populations for inclusion in the following generation's population. The non-dominated sorting discussed earlier for selection is used here to order the combined population of original and offspring solutions and then select the best solutions. The number of solutions to be selected is determined by population size.

The following pseudo-code summarizes the described algorithm steps proposed for joint modeling of variables and objectives.

```
Joint Variable-Objective Estimation of Distribution Algorithm:
P[0] = Generate an initial population
F[0] = Compute the objective values(P[0])
t = 0
While termination criteria are not satisfied do
   {S, G} = Select a subset of solutions(P[t], F[t])
   D = Form the extended solutions(S, G)
   // D is duly normalized and standardized
   C = Compute the regularized covariance matrix(D)
   M = Learn the GBN model(C)
   Q = Sample offspring from the model(M, G)
   H = Compute the objective values(Q)
   {P[t+1], F[t+1]} = Replace offspring in population(P[t], F[t], Q, H)
   t = t + 1
End while
```

3 Experiments

The proposed algorithm is tested on a number of test functions and the results are compared to those of other algorithms. To test the effect of including the objectives in the model building and sampling of the algorithm two versions of the algorithm are considered here. The first algorithm learns a joint model of both the objectives and variables, which will be called Joint GBN-EDA (JGBN-EDA). The other version does not consider objective information and only uses variables for model learning, very similar to the Estimation of Gaussian Network Algorithm (EGNA) [15], and will be called (normal) GBN-EDA in this paper.

The performance results are compared against two other algorithms: The Non-dominated Sorting Genetic Algorithm (NSGA-II) [7], considered as a reference algorithm in many multi-objective optimization studies, is based on traditional GA and uses special type of crossover and mutation methods to deal with real-valued strings; The Regularity-Model based Multi-objective Estimation of Distribution Algorithm (RM-MEDA) [33] assumes a certain type of smoothness for Pareto set and uses the Local Principal Component Analysis (LPCA) algorithm to build a piece-wise continuous manifold with a dimension equal to one less than the number of objectives.

The two algorithms use the non-dominated sorting method for selecting promising solutions. While both algorithms generate an offspring population of the size of the original population, only NSGA-II employs a replacement mechanism where the best solutions in both offspring and original populations are selected for the next generation. In RM-MEDA the newly generated offspring solutions completely replace the original population. All of the algorithms are implemented in Matlab®.

The implementation of JGBN-EDA is done using the Matlab® toolbox for EDAs [28]. The algorithm has been tested using both the non-dominated sorting and the fitness averaging for selection. However, the results obtained with the simpler fitness averaging were superior to those obtained with the other algorithm. Such a behavior is also reported by others [10]. Therefore the results presented here are those obtained with fitness averaging. The non-dominated sorting is used only in the replacement step of the algorithm as was discussed above.

To evaluate the non-dominated fronts obtained by each algorithm the Inverted Generational Distance (IGD) measure is used that accounts both for diversity and convergence to the true Pareto front at the same time. The measure is given by the following relation taken from [33]

$$IGD_{F^*}(F) = \left(\sum_{s \in F^*} \{ \min d(s, s'), \forall s' \in F \} \right) / |F^*| \tag{5}$$

where $d()$ gives the Euclidean distance between the two points and F^* is a uniform sampling of the true Pareto front of the problem. The statistical significance in the differences between the results is checked with the Kruskal-Wallis test. Kruskal-Wallis performs a non-parametric one-way ANOVA to accept or reject the null hypothesis that independent samples of two or more groups come from distributions with equal medians, and returns the p-value for that test. The test significance level is set to 0.05.

3.1 Test Functions

There are many test sets proposed in the literature for evaluating multi-objective optimization algorithms [8,9,11,24]. However not all of these functions exhibit real-world problem properties that can be used to evaluate different aspects of an algorithm. Features like scalability and non-separability both in the variable and objective spaces, multi-modality, biased or disconnected Pareto fronts that can pose significant challenges to any multi-objective optimization algorithm.

In this study, the Walking Fish Group (WFG) functions proposed by Huband et al. [11,12] are used for experiments. All of the functions, which are to be minimized, can be represented with the following relation

$$f_j(\mathbf{x}) = D \cdot x_m + S_j \cdot h_j(x_1, \ldots, x_{m-1}) \tag{6}$$

where D and S_j are scaling factors, h_j are shape functions (e.g. concave, convex, mixed, disconnected or degenerated) defined on $m - 1$ (m is the number of objectives) position variables, and x_m is the distance variable. Position and distance variables are obtained by applying different transformation function on the input variables. The test set includes 9 functions that all have dissimilar variable domains and different Pareto optimal trade-off magnitudes, have no extremal nor medial values for variables, and undergo complex transformations to create other properties like deceptiveness and many-to-one mappings.

3.2 Experimental Results

In the following experiments the algorithms are tested on WFG functions with 5, 10 and 20 objective functions. The number of objectives are related to the number of position and distance variables with two factors that determine the number of problem variables. Therefore to keep the computational costs of the experiments in an affordable level, the number of variables is set to 28, 50 and 50 respectively. The reported results for each algorithm on each test function are computed from 25 runs in the case of 5 and 10 objectives, and 10 runs for 20 objectives. For all algorithms the maximum number of function evaluations is set to $1.5 \cdot 10^5$, $3 \cdot 10^5$ and $7.5 \cdot 10^5$, and the population size to 600, 1000 and 1500 respectively for different number of objectives. Figure 3 shows the typical fronts obtained by the algorithms for some of the test functions (WFG2, WFG4, WFG6 and WFG7) with 3 objectives and 20 variables to give an idea of how the algorithms are performing for smaller number of objectives. The population size is set to 200 in this case.

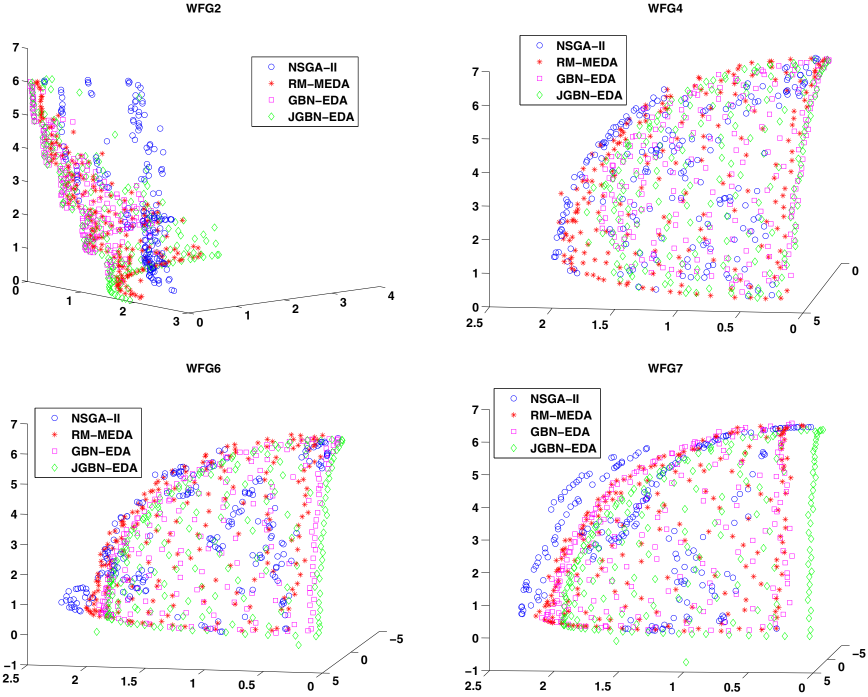


Fig. 3. Typical Pareto fronts obtained for different algorithms performance on some of (*WFG*) functions with 3 objectives

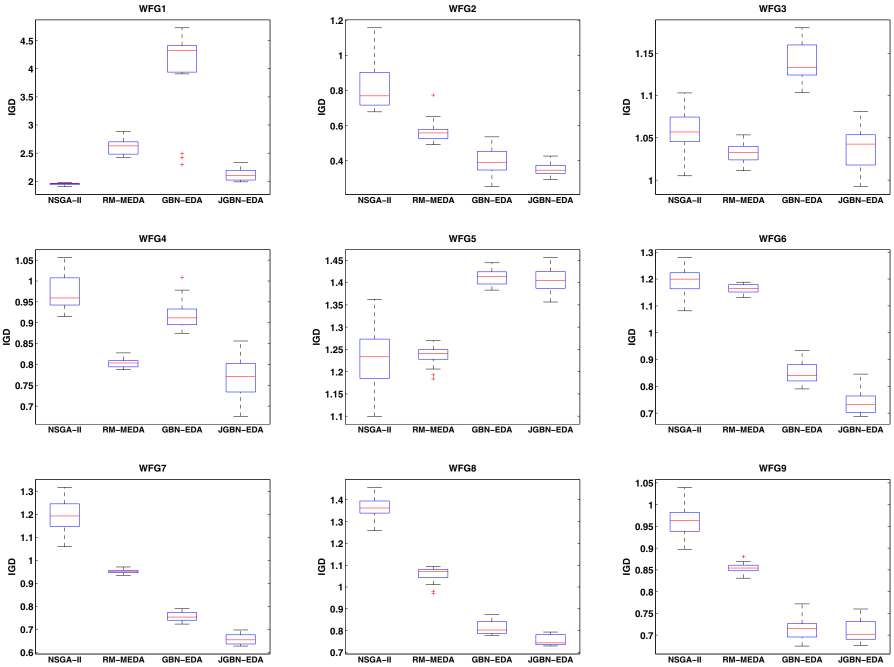


Fig. 4. Comparison of different algorithms performance on (*WFG*) functions with 5 objectives

Fig. 4 shows the results obtained for each of the algorithms for WFG functions with 5 objectives. As it can be seen in the figure, the incorporation of objectives in the modeling of JGBN-EDA enables this algorithm to obtain a (significantly) better performance on most of the functions. A direct comparison of GBN-EDA and JGBN-EDA shows the effectiveness of joint modeling where except for WFG5 and WFG9 functions, the latter algorithm can obtain significantly better results on all other functions ($p = 0.05$).

WFG5 is a separable and deceptive function while WFG9 is a non-separable, biased, multi-modal and deceptive function. According to the presented results, for deceptive functions the information provided by objectives does not have a great impact on the probabilistic model learnt in JGBN-EDA for generating better solutions. Also, the algorithm is not able to properly utilize the separability of variables in WFG5 to obtain better fronts. However, the results obtained for WFG9 shows that non-separability and multi-modality of this problem is completely addressed with the modeling used in GBN-EDAs which makes them able to dominate the other two competitor algorithms on this function.

As the number of objectives grows to 10 and 20 (Figs. 5 and 6), the proposed algorithm performance deteriorates in comparison to other algorithms. The other two algorithms also show a diverse behavior on different number of

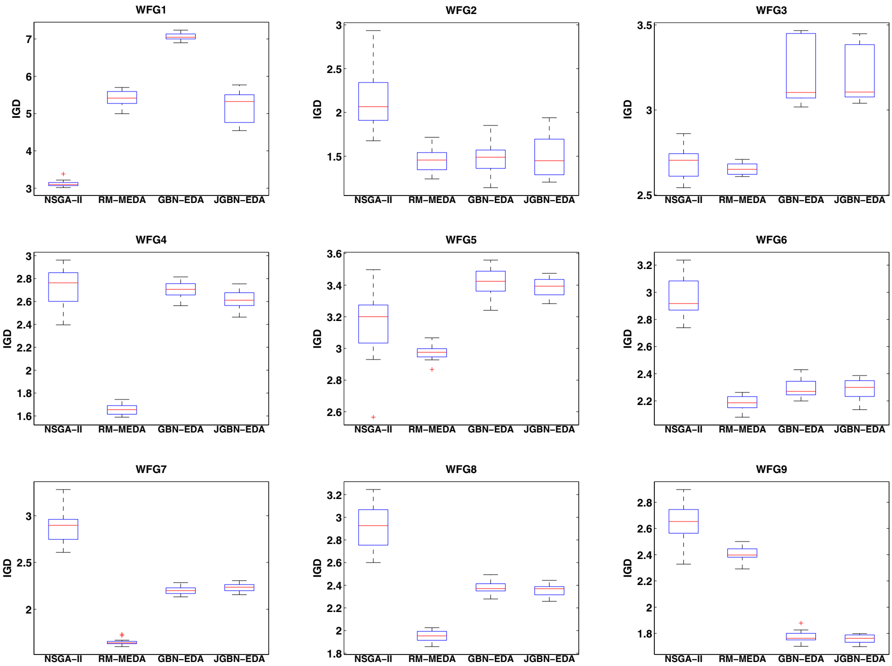


Fig. 5. Comparison of different algorithms performance on (*WFG*) functions with 10 objectives

objectives. While the fronts resulted by NSGA-II on 10 objectives functions are not comparative to other algorithms, it is able to find significantly better results for most of the functions with 20 objectives. RM-MEDA is showing an apposite behavior, being the better algorithm on the 10 objective case. For these high number of objectives the performance of the proposed JGBN-EDA seems to be less varying when compared on both 10 and 20 objectives WFG functions.

The mixed concave-convex front of WFG1 with some flat regions in the Pareto set is best addressed by the recombination method employed in NSGA-II. The inclusion of objective values in the modeling of JGBN-EDA has a major influence in improving the performance on this function when compared to GBN-EDA. On the other hand, the disconnected front of WFG2 causes a diverse performance of NSGA-II, while the EDAs are almost performing similar on 10 and 20 objectives functions. The significantly better performance of JGBN-EDA on WFG9 is repeated for 10 and 20 objectives.

It seems that when the number of objectives increase to a high number (e.g. 20 objectives), the modeling present in EDAs gets saturated for small populations and can not help the algorithm to make progress. In fact for such a large number of objectives small population sizes may not suffice to represent the Pareto front.

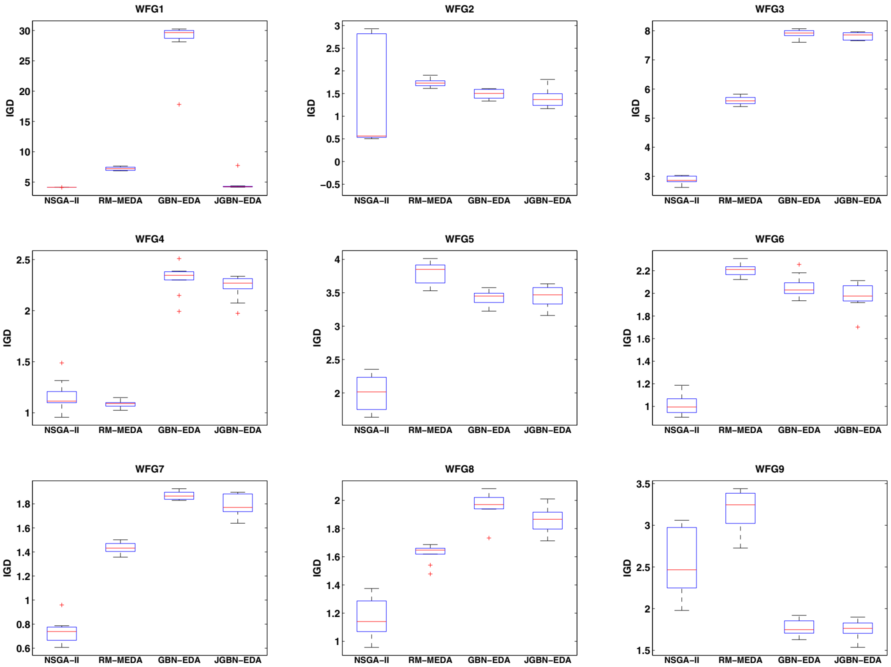


Fig. 6. Comparison of different algorithms performance on (*WFG*) functions with 20 objectives

This issue also poses a challenge for representing the true Pareto front to be used for the computation of performance measure. Although in this study front approximations with several hundred thousands of points have been used for this purpose, some areas of the front may not yet be covered properly.

4 Conclusion and Further Research

A joint modeling of objectives and variables was proposed to be used for multi-objective optimization in this paper. The proposed EDA, learns a Gaussian Bayesian network to encode the relationships between the variables and objectives, from a set of extended solutions formed by appending the objectives to the variables of each solution. The learnt probabilistic model will have both variable and objective nodes. The sampling procedure generates new values for problem variables using the objective values when such dependencies are encountered in the model.

The performance of the algorithm was tested on a number of multi-objective test functions with different real-world properties. The results show that the incorporation of objectives in the modeling can help the algorithm to obtain better fronts on some of the functions. The results were also compared against two other algorithms and indicated that the idea of including objective values in the modeling step of an EDA is promising and can help to obtain better fronts in multi-objective optimization.

The algorithm had some problems in detecting the correct search bias for some deceptive functions using the proposed joint modeling. Also for some of the functions, the algorithm was not able to obtain competitive fronts when applied to high number of objectives . The effect of proper population sizing on the performance of the algorithm specially for many objectives problems should be studied in more detail. Other ways of utilizing the objective values in the sampling process can be used to reduce this shortcoming.

Nevertheless the proposed algorithm can be seen as an alternative for using modeling in the multi-objective optimization. The information provided by dependencies between the objectives can be further investigated for obtaining the relationships in problems with many objectives. The factorization obtained by the explicit inclusion of objectives in the model is also another possibility to simplify the problem. A future line of research associated with this is to force some special kind of relations in the joint probabilistic model, like those discussed for class-bridge decomposable multi-dimensional Bayesian network classifiers [2], that will allow to decompose the problem into smaller subproblems, thus easing the learning and sampling tasks.

Acknowledgments. This work has been partially supported by TIN2010-20900-C04-04, Consolider Ingenio 2010-CSD2007-00018 and Cajal Blue Brain projects (Spanish Ministry of Science and Innovation).

References

1. Abraham, A., Jain, L., Goldberg, R. (eds.): Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. Advanced Information and Knowledge Processing. Springer, Berlin (2005)
2. Bielza, C., Li, G., Larrañaga, P.: Multi-dimensional classification with Bayesian networks. Technical Report UPM-FI/DIA/2010-1, Artificial Intelligence Department, Technical University of Madrid, Madrid, Spain (2010)
3. Brockhoff, D., Zitzler, E.: Dimensionality reduction in multiobjective optimization: The minimum objective subset problem. In: Waldmann, K.-H., Stocker, U.M. (eds.) Operations Research Proceedings 2006, pp. 423–429. Springer, Berlin (2007)
4. Brockhoff, D., Zitzler, E.: Objective reduction in evolutionary multiobjective optimization: Theory and applications. Evolutionary Computation 17(2), 135–166 (2009)
5. Buntine, W.: Theory refinement on Bayesian networks. In: Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence, vol. 91, pp. 52–60. Morgan Kaufmann, San Francisco (1991)
6. Coello, C.: An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1999), vol. 1, pp. 3–13 (1999)

7. 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)
8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002), vol. 1, pp. 825–830 (2002)
9. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 105–145. Springer, Heidelberg (2005)
10. Garza-Fabre, M., Pulido, G., Coello, C.: Ranking methods for many-objective optimization. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS, vol. 5845, pp. 633–645. Springer, Heidelberg (2009)
11. Huband, S., Barone, L., While, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 280–295. Springer, Heidelberg (2005)
12. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477–506 (2006)
13. Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., Nojima, Y.: Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO 2008), pp. 649–656. ACM, New York (2008)
14. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. In: Adaptive Computation and Machine Learning. The MIT Press, Cambridge (2009)
15. Larrañaga, P., Etxeberria, R., Lozano, J., Peña, J.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: Wu, A. (ed.) Proceedings of the 2000 Genetic and Evolutionary Computation Conference (GECCO 2000) Workshop Program, pp. 201–204. Morgan Kaufmann, San Francisco (2000)
16. Larrañaga, P., Lozano, J. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2001)
17. Lauritzen, S.L.: Propagation of probabilities, means, and variances in mixed graphical association models. Journal of the American Statistical Association 87(420), 1098–1108 (1992)
18. López Jaimes, A., Coello Coello, C.A., Chakraborty, D.: Objective reduction using a feature selection technique. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO 2008), pp. 673–680. ACM, New York (2008)
19. Lozano, J., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.): Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. Studies in Fuzziness and Soft Computing, vol. 192. Springer, Heidelberg (2006)
20. Martí, L., Garcia, J., Antonio, B., Coello, C.A., Molina, J.: On current model-building methods for multi-objective estimation of distribution algorithms: Shortcommings and directions for improvement. Technical Report GIAA2010E001, Department of Informatics, Universidad Carlos III de Madrid, Madrid, Spain (2010)
21. Miquélez, T., Bengoetxea, E., Larrañaga, P.: Evolutionary computation based on Bayesian classifiers. International Journal of Applied Mathematics and Computer Science 14(3), 335–350 (2004)

22. Miquélez, T., Bengoetxea, E., Larrañaga, P.: Evolutionary Bayesian Classifier-Based Optimization in Continuous Domains. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 529–536. Springer, Heidelberg (2006)
23. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions i. binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
24. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J., Tino, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 792–802. Springer, Heidelberg (2004)
25. Pelikan, M., Sastry, K., Cantú-Paz, E. (eds.): Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications. SCI. Springer, Heidelberg (2006)
26. Pelikan, M., Sastry, K., Goldberg, D.: Multiobjective estimation of distribution algorithms. In: Pelikan, et al. (eds.) [25], pp. 223–248
27. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. In: Adaptive Computation and Machine Learning. The MIT Press, Cambridge (2005)
28. Santana, R., Bielza, C., Larrañaga, P., Lozano, J.A., Echegoyen, C., Mendiburu, A., Armañanzas, R., Shakya, S.: Mateda-2.0: Estimation of distribution algorithms in MATLAB. Journal of Statistical Software 35(7), 1–30 (2010)
29. Sbalzarini, I., Mueller, S., Koumoutsakos, P.: Multiobjective optimization using evolutionary algorithms. In: Proceedings of the 2000 Summer Program of Studying Turbulence Using Numerical Simulation Databases–VIII, vol. 1, pp. 63–74. Center for Turbulence Research (November 2000)
30. Schäfer, J., Strimmer, K.: A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical Applications in Genetics and Molecular Biology 4(1) (2005)
31. Schmidt, M., Niculescu-Mizil, A., Murphy, K.: Learning graphical model structure using L1-regularization paths. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007), vol. 2, pp. 1278–1283. AAAI Press, Menlo Park (2007)
32. Thierens, D., Bosman, P.: Multi-objective mixture-based iterated density estimation evolutionary algorithms. 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), pp. 663–670. Morgan Kaufmann, San Francisco (2001)
33. Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 12(1), 41–63 (2008)
34. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)

A Bi-objective Based Hybrid Evolutionary-Classical Algorithm for Handling Equality Constraints

Rituparna Datta and Kalyanmoy Deb

Kanpur Genetic Algorithms Laboratory (KanGAL),
Department of Mechanical Engineering,
Indian Institute of Technology Kanpur, PIN 208016, India
{rdatta,deb}@iitk.ac.in
http://www.iitk.ac.in/kangal

Abstract. Equality constraints are difficult to handle by any optimization algorithm, including evolutionary methods. Much of the existing studies have concentrated on handling inequality constraints. Such methods may or may not work well in handling equality constraints. The presence of equality constraints in an optimization problem decreases the feasible region significantly. In this paper, we borrow our existing hybrid evolutionary-cum-classical approach developed for inequality constraints and modify it to be suitable for handling equality constraints. This modified hybrid approach uses an evolutionary multi-objective optimization (EMO) algorithm to find a trade-off frontier in terms of minimizing the objective function and the constraint violation. A suitable penalty parameter is obtained from the frontier and then used to form a penalized objective function. The procedure is repeated after a few generations for the hybrid procedure to adaptively find the constrained minimum. Unlike other equality constraint handling methods, our proposed procedure does not require the equality constraints to be transformed into an inequality constraint. We validate the efficiency of our method on six problems with only equality constraints and two problems with mixed equality and inequality constraints.

Keywords: Evolutionary multi-objective optimization, Constraint handling, Equality constraint, Penalty Function, Bi-Objective optimization, Hybrid methodology.

1 Introduction

Multi-objective optimization using evolutionary algorithms (EAs) is currently one of the fast growing research area in science and different disciplines of engineering. Most practical optimization problems usually consists a large number of non-linear, non-convex, and discontinuous constraint as well as objective functions. A typical single objective constrained optimization problem is given below:

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 313–327, 2011.

$$\begin{aligned} \text{Minimize } & f(\boldsymbol{x}), \\ \text{Subject to } & g_j(\boldsymbol{x}) \geq 0, j = 1, \ldots, J, \\ & h_k(\boldsymbol{x}) = 0, k = 1, \ldots, K, \\ & x_i^l \leq x_i \leq x_i^u, i = 1, \ldots, n, \end{aligned} \tag{1}$$

where n is the number of variables, J inequality constraints, and K equality constraints. The function $f(\boldsymbol{x})$ is the objective function, $g_j(\boldsymbol{x})$ is the j-th inequality constraint and $h_k(\boldsymbol{x})$ is the k-th equality constraint. The i-th variable varies in the range $[x_i^l, x_i^u]$. An inequality constraint with zero value at the optimum is called an active constraint. However, the optimum solution must satisfy all equality constraints, thereby making each equality constraint an active constraint. Equality constraints are most difficult to handle using any optimization algorithm including an EA, since the optimum solution must lie on the intersection of all equality constraints. Due to this reason, the existence of equality constraints in a problem decreases the proportion of the feasible region in the space of all solutions remarkably and eventually the problem becomes so complicated that, finding a single feasible solution (as well as optimal solution) becomes a difficult task.

Penalty function method has proven it as one of the popular constraint-handling technique due to its simple principle and easy implementation requirements. However, one of the drawbacks of this approach is that it requires a proper value of the penalty parameter [1,2]. In the penalty function method, the original constrained optimization problem is replaced by an equivalent unconstrained problem with the inclusion of a penalty term which worsens the objective function value by adding a penalty proportional to the constraint violation (CV). If the value of the penalty parameter is too small, the constraints may be violated and the resulting solution may be infeasible. Similarly, if the value of penalty parameter is high, the solution may ignore the contribution from the objective function and the resulting solution may get stuck to a suboptimal solution.

Richardson [3] proposed guidelines for setting the penalty parameter in a genetic algorithm (GA). Gen et al. [4] proposed a tutorial survey of recent works on penalty techniques used in GAs up to 1996. All these earlier studies demanded an appropriate penalty parameter to be set adaptively or by the user for the method to work. In 2000, the second author suggested a penalty-parameter-less approach by simply comparing two solutions based on their feasibility and constraint violation values [5]. Coello [6] proposed a self-adaptive penalty function approach by using a co-evolutionary framework to adaptively find the penalty parameters.

Recent studies in constraint handling have focused in using a multi-objective optimization technique. In addition to the original objective function ($f(\boldsymbol{x})$), the constraint violation is included as an additional objective function thereby making the problem a bi-objective one. The bi-objective problem is then solved using an EMO technique. Surry et al. [7] suggested a multi-objective based constraint handling (COMOGA), where the population was first ranked based on the constraint violation value and then ranked using the objective function value. Camponogara et al. [8] proposed a similar bi-objective idea, in which a search direction is generated based on a domination check of two solutions from the population. Angantyr [9] proposed a constraint handling based on multi-objective real-coded Genetic Algorithm (GA) inspired by the penalty approach. Coello [10] suggested the violation of each constraint as a different objective. The bi-objective idea is applied near the currently best population member in an adaptive manner using a reference-based EMO [11]. Since the task is to find the constrained minimum of the original problem, this modified search allowed a quicker discovery of the optimum. Echeverri et al. [12] proposed a bi-objective based bi-phase method in which, the objective function is not considered in the first phase and the search is directed towards finding a single feasible solution followed by solving the two objective problem.

Recent studies on evolutionary constraint handling methods have also concentrated on using a hybrid methodology in which an EA is coupled with a classical optimization algorithm in order to develop a much better algorithm than each method alone. Some such methods can be found elsewhere [11,13,14,15].

As mentioned above, not much emphasis has been made in handling equality constraints explicitly. A usual way to handle equality constraints in EAs is to transform an equality constraint into an inequality constraint in the following way:

$$\epsilon - |h_k(\boldsymbol{x})| \geq 0, \tag{2}$$

where ϵ is the tolerance value by which the constraint can be violated during the optimization process. This approach requires a suitable value of the parameter ϵ. If a large value is taken, the optimization process would be easier but the resulting solution may not be a feasible one with respect to the original problem. On the other hand, if a small value is chosen, the optimization task to find a feasible solution would be difficult. Hinterding [16] handled equality constraints in which the tolerance value (ϵ) mentioned above was reduced in predefined manner with the generation counter. They manipulated the equality constraints to replace some variables and this process has reduced the problem size. However, the strategy may not be applicable in general, particularly where variable elimination from the equality constraints may be a difficult task. Peconick et al. [17] proposed CQA-MEC (Constraint Quadratic Approximation for Multiple Equality Constraints) which approximated the non-linear constraints via quadratic functions. Some other studies can be found elsewhere [18,19].

Recently, we have suggested a constraint handling methodology based on a bi-objective evolutionary optimization strategy [20] to estimate the penalty parameter for a problem from the obtained two-objective Pareto-optimal front. Thereafter, an appropriate penalized function is formed and a classical local search method is used to solve the problem. This method was developed for solving inequality constraints. In this work, we extend this methodology for handling equality constraints. The newly proposed methodology uses a different penalty parameter update strategy which seems to be not require any tolerance value to be set, as it is the case with most equality constraint handling EAs.

In the remainder of this paper, we describe the modified bi-objective hybrid methodology. Thereafter, we present simulation results on eight different test problems taken from the existing literature. Finally, based on the obtained result, conclusions are made.

2 Bi-objective Hybrid Equality Constraint Handling Method

The details of the bi-objective hybrid method, combining a bi-objective EA and the penalty function approach for inequality constraints, are described in our earlier study [20]. Here, we discuss the difficulties of directly using the previous procedure and suggest modifications to suit the hybrid algorithm for handling equality constraints.

The presence of equality constraints in an optimization problem usually shrinks the feasible space significantly and makes the problem more complicated to find a feasible solution. It is clear that a feasible solution must now lie on the boundary of each equality constraint. The penalty function approach aggregates the constraint violation, arising from each constraint. Since it is difficult to satisfy a non-linear equality constraint easily, most solutions will correspond to certain constraint violation value. Since most constraints are unsatisfied, an aggregate of constraint violation values may not compare two solutions properly and emphasize the right solution. Since the aggregate CV is not a proper metric, the bi-objective problem of minimizing the objective function and minimizing CV is also not free from the same problem. Thus, the overall bi-objective hybrid approach described earlier may not perform well on equality constraints.

Here, we modify our earlier approach in two different ways. First, to emphasize constraint satisfaction, we increase the penalty parameter as soon as the local search method fails to find a feasible solution. For the local search algorithm, the penalty parameter value is increased by such an amount so as to guarantee that constraint violation term is given more emphasis than the objective function term, at least at the current best solution. For the EMO algorithm, the obtained R from the bi-objective frontier is also doubled to make sure constraint violation is given more emphasis. Second, since feasible solutions are difficult to find for equality constraints, we relax the constraint used in the bi-objective problem formulation. Instead of declaring a solution to be feasible to the bi-objective problem if the overall constraint violation of 0.2 times the number of constraints, we use 0.4 times the number of constraints. These two considerations should allow an adequate number of solutions to be considered and emphasized in both the bi-objective algorithm and also in the local search procedure.

2.1 Proposed Algorithm

This section describes the proposed algorithm based on the idea of principles of bi-objective handling of a constrained optimization and then use of penalty function method. First, the generation counter is set at $t = 0$ and $R_{new} = 0$.

Step 1: An evolutionary multi-objective optimization (EMO) algorithm (NSGA-II [21]) is applied to the following bi-objective optimization problem to find the non-dominated trade-off frontier:

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}), \\ &\text{minimize } \text{CV}(\boldsymbol{x}), \\ &\text{subject to } \text{CV}(\boldsymbol{x}) \leq c, \\ &\qquad x^{(L)} \leq x \leq x^{(U)}. \end{aligned} \tag{3}$$

The function $\text{CV}(\boldsymbol{x})$ is defined as follows:

$$\text{CV}(\boldsymbol{x}) = \sum_{i=1}^{J} \langle \hat{g}_j(\boldsymbol{x}) \rangle + \sum_{k=1}^{K} \left| \hat{h}_k(\boldsymbol{x}) \right|. \tag{4}$$

The constraints $\hat{g}_j$ and $\hat{h}_k$ are normalized functions of inequality and equality constraints, respectively [1,20]. The operator $\langle \cdot \rangle$ denotes the bracket operator. We have used $c = 0.4(J + K)$ in all our simulations.

Step 2: If $t > 0$ and $((t \bmod \tau) = 0)$, the penalty function approach is used, otherwise Step 1 is re-executed by incrementing t by one (here τ is the frequency of local search). First, we compute the penalty parameter R from the current non-dominated front as follows. Like in our earlier approach [20], when at least four solutions satisfying $\text{CV}(\boldsymbol{x}) \leq 0.4(J + K)$ are present in the population, a cubic polynomial ($f = a + b(CV) + c(CV)^2 + d(CV)^3$) is fitted for the obtained non-dominated points and the penalty parameter is estimated by finding the slope at CV=0. This gives $R = -b$. Since this is a lower bound on R, we use twice this value and set $R \leftarrow (-2b + R_{new})$.

Step 3: Thereafter, the following local search problem is solved using the current-best EMO solution (in terms of CV) as the initial point:

$$\begin{aligned} &\text{minimize } P(\boldsymbol{x}) = f(\boldsymbol{x}) + R \left[\textstyle\sum_{j=1}^{J} \langle \hat{g}_j(x) \rangle^2 + \sum_{k=1}^{K} \left(\hat{h}_k(x) \right)^2 \right], \\ &\qquad x^{(L)} \leq x \leq x^{(U)}. \end{aligned} \tag{5}$$

Say, the solution to the above problem is $\boldsymbol{x}^{LS}$.

Step 4: If $\boldsymbol{x}^{LS}$ is feasible (constraint violation $\leq \delta_f$), we check whether the difference between $f(\boldsymbol{x}^{LS})$ and the objective value of the previous local searched solution is smaller than a user-defined parameter δ_f. If not, the algorithm proceeds to Step 1 by incrementing t by one. Else, the algorithm is terminated, and $\boldsymbol{x}^{LS}$ is declared as the optimized solution.

If $\boldsymbol{x}^{LS}$ is infeasible, we proceed to Step 5.

Step 5: We change the penalty parameter value for the next local search algorithm, as follows:

$$R_{new} = R + \frac{f(\boldsymbol{x}^{LS})}{CV(\boldsymbol{x}^{LS})}, \tag{6}$$

We increment t by one and then proceed to Step 1.

There are two parameters to the above algorithm: τ and δ_f. By performing some experimental studies on standard test problems, we have set $\tau = 5$ and $\delta_f = 10^{-4}$ in all our simulations.

It is also interesting to note that, the penalty parameter R is no more a user-tunable parameter and gets learnt adaptively from the obtained non-dominated front as well as infeasible local searched solutions. We use Matlab's `fmincon()` procedure to solve the penalized function (Equation 5). Another interesting aspect is that we have not used any tolerance value ϵ to fix the equality constraints here.

3 Simulation Results on Standard Test Problems

We now apply our proposed methodology to eight constrained test problems, taken from [22]. Following parameter values are used: population size $= 20n$ (n is the number of variables), SBX probability$= 0.9$, SBX index $= 10$, polynomial mutation probability $= 1/n$, and mutation index $= 100$. The description of SBX and polynomial mutation operations can be found elsewhere [21]. The termination criterion is described in Section 2.1. In each case, we run our algorithm 50 times from different initial populations.

3.1 Problem g03

The problem has 10 variables and a single equality constraint [22]. The constraint is not normalized. Figure 1 indicates the evolution of the objective function value of the population, eventually to the best and the corresponding constraint violation value with the generation for a typical simulation out of 50 initial populations. The figure shows that initially solutions are infeasible up to fourth generation. At generation 5, the first local search is performed, as at least four solutions satisfying the constraint in Equation 3 are found. After 10-th generation when the second local search is done, the approach reaches near the optima. At 15-th generation, two consecutive values of local searched solution are close to each other and the algorithm is terminated. The figure also shows the change of CV from an positive value (infeasible) to zero (feasible). Table 1 presents the function evaluations required by our approach to terminate according to the termination condition described in the algorithm.

However, to compare our approach with existing studies, we rerun our algorithm with a different termination condition. In Step 4, when the objective function value of the feasible $\boldsymbol{x}^{LS}$ is close to the best-reported function value (within 10^{-4}), the algorithm is terminated. Table 2 tabulates and compares the overall function evaluations needed by our approach with three other methodologies taken from literature.

Table 1. Comparison of obtained solutions found out of 50 runs with best-known optimal solutions . NSGA-II and local search function evaluations (FEs) are shown separately. Since the algorithm is terminated when two consecutive local searches produce similar solutions, in some cases (as in problem g17) the smallest FE solution need not best in terms of function value.

Problem	Best known optima	Obtained feasible values		
		Best	Median	Worst
g03 (FE)		**3,813**	**4,435**	**11,920**
NSGA-II + Local		3,000 + 813	3,000 + 1,435	9,000 + 2,920
(f^*)	**−1.000500**	**−1.000350**	−1.000349	−1.001999
g05 (FE)		**9,943**	**11,994**	**14,539**
NSGA-II + Local		8,400 + 1,543	10,000 + 1,994	12,400 + 2,139
(f^*)	**5126.496714**	**5125.931709**	5126.338978	5126.336735
g11 (FE)		**1,334**	**1,559**	**1,612**
NSGA-II + Local		1,200 + 134	1,400 + 159	1,400 + 212
(f^*)	**0.749900**	**0.749534**	0.749776	0.749758
g13 (FE)		**1,499**	**2,577**	**3,778**
NSGA-II + Local		1,000 + 499	2,000 + 577	3,000 + 778
(f^*)	**0.053941**	**0.0539169458**	0.0539899948	0.0539162638
g14 (FE)		**10,498**	**12,720**	**13,692**
NSGA-II + Local		9,000 + 1,498	11,000 + 1,720	12,000 + 1,692
(f^*)	**−47.764888**	**-47.762282**	-47.761435	-47.761438
g15 (FE)		**1,431**	**2,254**	**3,700**
NSGA-II + Local		1,200 + 231	1,800 + 454	2,100 + 1,600
(f^*)	**961.715022**	**961.715195**	961.715403	961.735327
g17 (FE)		**2,109**	**4,344**	**13,406**
NSGA-II + Local		1,800 + 309	3,000 + 1,344	7,200 + 6,206
(f^*)	**8853.539674**	8927.602048	**8853.537314**	8853.748783
g21 (FE)		**4,044**	**5,289**	**9,456**
NSGA-II + Local		3,500 + 544	4,200 + 1,089	8,400 + 1,056
(f^*)	**193.724510**	**193.775400**	193.778862	193.781075

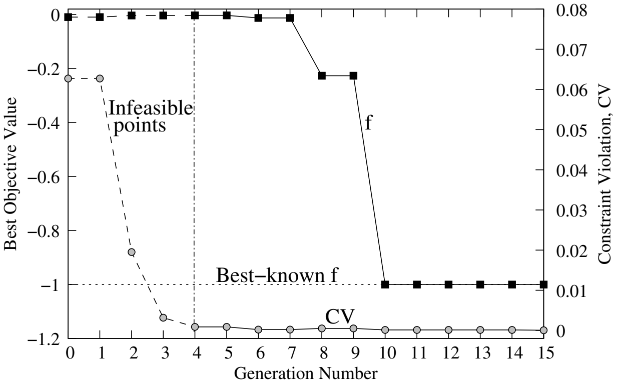


Fig. 1. Function value reduces with generation for g03 problem

Table 2. Comparison of function evaluations needed by the proposed approach and the existing 3 approach [23], [24], [25]. NSGA-II and local search evaluations are shown separately. The best ones are shown in bold. In most of the problems, the hybrid approach outperforms the existing approach.

Problem	Zavala, Aguirre & Diharce [23]			Takahama & Sakai [24]		
	Best	Median	Worst	Best	Median	Worst
g03	97,892	1,06,180	1,22,540	30,733	35,470	41,716
g05	1,49,493	1,65,915	1,88,040	15,402	16,522	**17,238**
g11	89,734	1,12,467	1,27,650	4,569	4,569	4,569
g13	1,49,727	1,60,964	1,68,800	2,707	4,918	11,759
g14	1,38,471	1,49,104	1,65,292	30,925	32,172	32,938
g15	1,27,670	1,35,323	1,47,268	4,053	6,805	**10,880**
g17	2,21,036	2,32,612	2,36,434	15,913	16,511	**16,934**
g21	2,06,559	2,21,373	2,33,325	31,620	35,293	35,797

Problem	Brest [25]			Proposed Hybrid Approach		
	Best	Median	Worst	Best	Median	Worst
g03	1,84,568	2,15,694	2,54,105	**4,687**	**5,984**	**33,336**
g05	49,765	53,773	57,863	**10,048**	**11,101**	25,671
g11	52,128	83,442	1,05,093	**1,334**	**1,559**	**1,612**
g13	1,38,630	1,47,330	4,28,869	**1,499**	**2,577**	**3,778**
g14	2,23,822	2,42,265	2,56,523	**7,042**	**9,265**	**11,449**
g15	1,53,943	1,57,822	1,60,014	**1,082**	**2,117**	22,772
g17	1,85,888	2,05,132	2,55,333	**2,728**	**4,638**	2,33,239
g21	1,31,557	1,49,672	1,58,079	**2,342**	**3,392**	**7,062**

3.2 Problem g05

The problem has five variables with five constraints, of which three constraints are of equality type and are multi-modal in nature [22]. The constraints are normalized as follows:

$$\begin{aligned}
\hat{g}_1(\boldsymbol{x}) &= \left[x4 - x3 + 0.55\right]/0.55 \geq 0,\\
\hat{g}_2(\boldsymbol{x}) &= \left[x3 - x4 + 0.55\right]/0.55 \geq 0,\\
\hat{h}_3(\boldsymbol{x}) &= \left[1000\sin(-x3 - 0.25) + 1000\sin(-x4 - 0.25) + 894.8 - x1\right]/1000 = 0,\\
\hat{h}_4(\boldsymbol{x}) &= \left[1000\sin(x3 - 0.25) + 1000\sin(x3 - x4 - 0.25) + 894.8 - x2\right]/1000 = 0,\\
\hat{h}_5(\boldsymbol{x}) &= \left[1000\sin(x4 - 0.25) + 1000\sin(x4 - x3 - 0.25) + 1294.8\right]/1000 = 0.
\end{aligned}$$

Figure 2 shows that up to 90 generations, no feasible solution to the original problem is found. This is due to the existence of equality constraints in the problem. We already discussed that equality constraints shrinks the feasible region and finding feasible solution is very difficult. The figure also shows the variation in the population-best objective value for a particular run. At generation 5, at least four solutions satisfying Equation 3 are found and the local search is executed. It helps reduce the objective value, however another 17 local searches are needed to get close to the constrained minimum solution. Since the variation in function values between two consecutive local searches (generations 120 and 125) is within 10^{-4}, the algorithm terminates then. The objective function value at a generation is joined with a solid line with the next generation value if the solution is feasible at the current generation, otherwise a dashed line is used. Table 1 shows the best, median and worst function evaluations with corresponding objective function value obtained using the proposed approach.

The problem is also solved using a termination criterion depending on closeness to the best-reported solution. When two objective function value from the local search is feasible and the difference is within 10^{-3}, the algorithm terminates. Table 2 indicates that in terms of best and median function evaluations our approach is better than all others. If we compare in terms of worst performance, Takahama et al. [24] found better solutions compared to other algorithms.

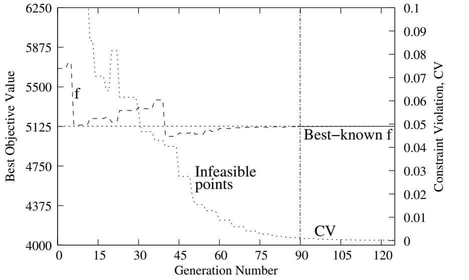


Fig. 2. Function value reduces with generation for g05 problem

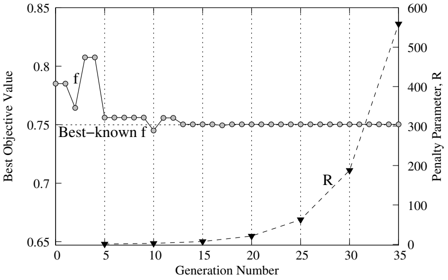


Fig. 3. Function value reduces with generation for g11 problem

3.3 Problem g11

This problem has a single equality constraint [22], hence no normalization of the constraint is needed. Figure 3 indicates the difference in the population-best objective value for a typical run out of 50 runs. In this problem. all solutions are feasible right from the initial population. Here, we show the adaptation of penalty parameter R with generations. The penalty parameter value increases with generation, as it learns to adapt R every time the local search fails to find a feasible solution. The local search starts at generation 5 and the algorithm took 7 local searches to converge. After the third local search operation, at generation 15, the solution comes close to the best-reported constrained minimum solution. When the difference between two consecutive local searched solutions is in the order of 10^{-4}, the algorithm terminates. In this problem, the corresponding value of R is 589.95. The best performance of our algorithm needs only 1,334 solution evaluations, whereas the best-reported EA requires more than three times more function evaluations to achieve a similar solution. Tables 1 and 2 indicate the same for the problem.

3.4 Problem g13

This problem has five variables and three equality constraints [22]. The objective function has an exponential term. Since all constraints are of equality type, they are not normalized.

Figure 4 shows the variation in the objective function value of the population-best solution and the adaptation of penalty parameter with the increasing number of generations. The first local search starts at generation 5 and the algorithm take three more local searches to fulfill our termination criteria. Table 3 and Figure 4 show that starting with a lower value of R, the hybrid approach increases with generation number to a suitable value needed to find the constrained minima. Tables 1 and 2 present the results.

Table 3. Adaptation of penalty parameter values for problem g13

Gen.	R
0	0.0000
5	2.6255
10	6.8048
15	17.6183
20	45.2747

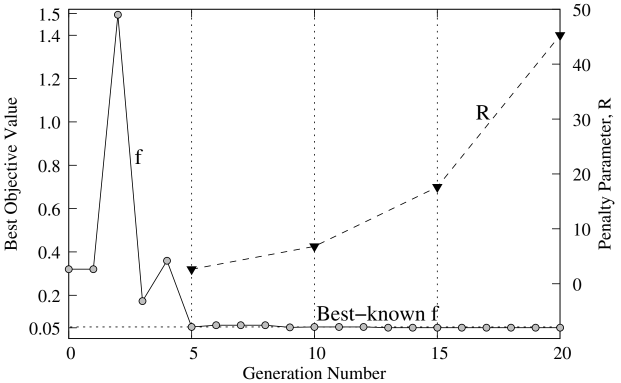


Fig. 4. Function value reduces with generation for g13 problem

3.5 Problem g14

This problem has a non-linear objective function with 10 variables and three equality constraints [22]. Constraints are not normalized. Table 1 shows the function evaluations and obtained best, median and worst objective function values using the proposed approach. For a particular simulation (out of 50 different initial populations), no feasible solution with respect to the original problem is found up to generation 39. However, the first local search starts after 20 generations, due to the first-time availability of at least four solutions satisfying the constraint in Equation 3. Thereafter, the proposed procedure takes a few more local searches to converge close to the best-reported solution. The value of the penalty parameter R at the final generation for this problem is found to be $297,811.12$. Figure 5 shows the evolution of the objective function value of the population from a lower value (with a constraint violation) to a higher value and thereafter converges close to the best-known optimum.

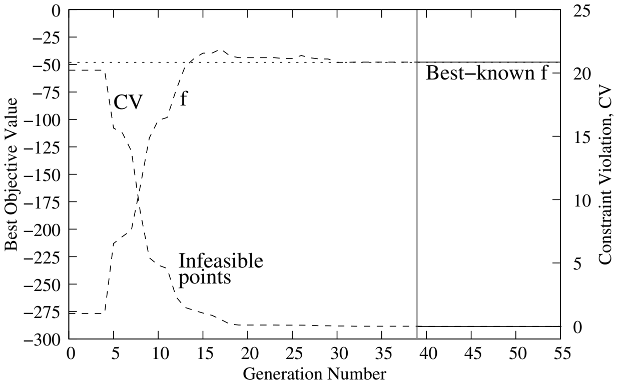


Fig. 5. Function value reduces with generation for g14 problem

Table 1 compares our best, median and worst solutions with corresponding function evaluations. Table 2 shows the FEs when the termination happens when a solution is found close to the best-known solution. The proposed approach is much faster than the existing methods.

3.6 Problem g15

This problem has only three variables and two non-linear equality constraints [22]. Constraints are not normalized. Figure 6 denotes the variation in the population-best objective value with number of generations. During the first 24 generations, the population can not find any feasible solution with respect to the original problem. However, at least four feasible solutions with respect to problem 3 are found at generation 20 and the first local search takes place. The local search helps to reduce the objective value. Thereafter, at 30-th generation, since a better solution (with a difference in objective value smaller than 10^{-4}) is not found, the algorithm terminates.

Table 2 shows the efficacy of the hybrid approach by using problem information as termination criteria. Again, the best performance of our approach (with function evaluations of 1,082) is more than 3.7 times better than that of the best-known approach (4,053) and our approach is also better in terms of median and worst-case performance compared to existing algorithms.

3.7 Problem g17

This problem has six variables and four equality constraints [22]. Constraints are not not normalized. All the constraints are multi-modal in nature, thereby making this problem difficult to solve. Table 1 shows that the best run could not reach the optima. In terms of median and worst performances, our proposed

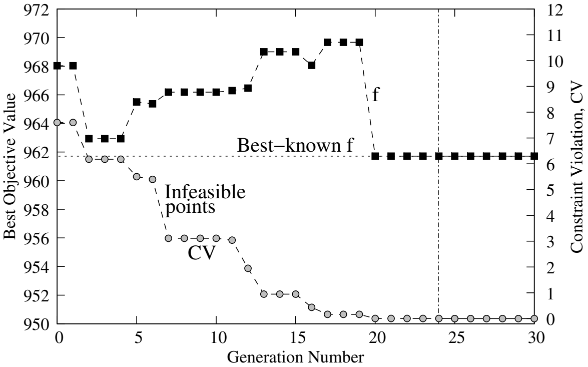


Fig. 6. Function value reduces with generation for g15 problem

approach is able to match the FEs of the existing algorithms. The algorithm is tested with 50 initial populations, of which 18 times it is not able to find the optima correctly with our progressive termination criterion. When a termination is checked to compare a solution's closeness to the best-known algorithm, our approach could not find the optima 12 times. In terms of the worst performance, the performance of the proposed approach is slightly worse. Figure 7 shows the evolution of the objective function value of the population-best solution and the corresponding constraint violation value with generation for a typical simulation out of 50 runs. The figure shows that no feasible solution is found up to 19 generations. However, at generation 10, the first local search is executed. With successive local searches of solving penalized functions with an adaptive R, the obtained solution get better and better with generations. At generation 25, the value of penalty parameter R is found to be $2,730,778.23$.

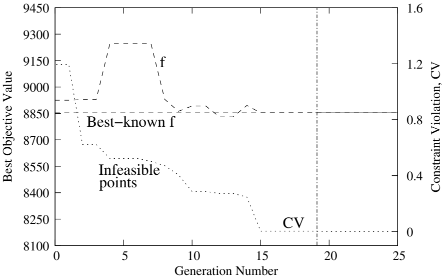


Fig. 7. Function value reduces with generation for g17 problem

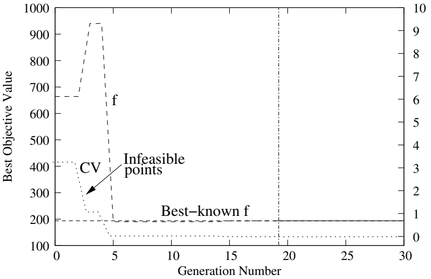


Fig. 8. Function value reduces with generation for g21 problem

3.8 Problem g21

The final problem has a combination of inequality and equality constraints [22]. Total number of constraints are six, out of which five are of equality type. The constraints are normalized as follows:

$$
\begin{aligned}
\hat{g}_1(\boldsymbol{x}) &= \left[x1 - 35x_2^{0.6} - 35x_3^{0.6}\right]/35 \geq 0,\\
\hat{h}1(\boldsymbol{x}) &= \left[-300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4\right]/7500 = 0,\\
\hat{h}_2(\boldsymbol{x}) &= \ 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 \ = 0,\\
\hat{h}_3(\boldsymbol{x}) &= \ -x_5 + \ln(-x_4 + 900) \ = 0,\\
\hat{h}_4(\boldsymbol{x}) &= \ -x_6 + \ln(x_4 + 300) \ = 0,\\
\hat{h}_5(\boldsymbol{x}) &= \ -x_7 + \ln(-2x_4 + 700) \ = 0.
\end{aligned}
$$

Figure 8 indicates that up to 19 generations, no feasible solution is found. This is due to the existence of a large number of equality constraints which make the problem complex. The first local search takes place at generation 5 and a solution having a similar objective value to that of the optimum solution is found, but all constraints are still not satisfied. To make the constraints satisfied with an order of 10^{-4}, it takes another five local searches.

Table 1 shows number of function evaluations required by our approach to terminate according to the termination condition described in the algorithm. In Step 4, when the objective function value of the feasible $\boldsymbol{x}^{LS}$ is close to the best-reported function value (within 10^{-3}), the algorithm is terminated. Table 2 also compares the overall function evaluations needed by our approach with three other methodologies taken from literature with a different termination criterion.

4 Conclusions

In this paper, we have developed a multi-objective based constraint handling methodology in which a bi-objective idea is coupled with classical penalty function approach in handling equality constraints and/or inequality constraints. The difficulty in getting an accurate convergence by an EMO procedure is balanced by the use of a local search method which involves a penalty-based classical optimization procedure. Moreover, the difficulty of the penalty function approach is alleviated by estimating the penalty parameter value adaptively by the trade-off points obtained by the EMO procedure. If the solution from the local search is infeasible, the penalty parameter is adjusted in a way to emphasize creation of more feasible solutions. The results on six equality and two combined inequality and equality constrained test problems are compared with three state-of-the-art algorithms and results indicate that our proposed procedure is robust (consistently better over multiple runs) and accurate (close to best-reported results). Further, our approach is faster in terms of function evaluations than the existing ones. In some cases, our approach is an order of magnitude better than the best-reported algorithm.

As a part of future work, we plan to pursue an adaptive normalization of constraints. In this present work, constraints have been normalized by using constant terms taken the constraint expressions, however an adaptive normalization strategy would be problem-independent. A parametric study on parameters τ and δ_f can be another extension to this work. The same idea can also be implemented for multi-objective constraint handling task.

Acknowledgment. The study is funded by Department of Science and Technology, Government of India under SERC-Engineering Sciences scheme (No. SR/S3/MERC/091/2009).

References

1. Deb, K.: Optimization for Engineering Design: Algorithms and Examples. Prentice-Hall, New Delhi (1995)
2. Reklaitis, G.V., Ravindran, A., Ragsdell, K.M.: Engineering Optimization Methods and Applications. Wiley, New York (1983)
3. Richardson, J.T., Palmer, M.R., Liepins, G.E., Hilliard, M.R.: Some guidelines for genetic algorithms with penalty functions. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 191–197. Morgan Kaufmann Publishers Inc., San Francisco (1989)
4. Gen, M., Cheng, R.: A survey of penalty techniques in genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation. IEEE Press, Los Alamitos (1996)
5. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)
6. Coello, C., Carlos, A.: Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry 41(2), 113–127 (2000)
7. Surry, P.D., Radcliffe, N.J., Boyd, I.D.: A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. In: Fogarty, T.C. (ed.) AISB-WS 1995. LNCS, vol. 993, pp. 166–180. Springer, Heidelberg (1995)
8. Camponogara, E., Talukdar, S.N.: A genetic algorithm for constrained and multi-objective optimization. In: 3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA), pp. 49–62 (1997)
9. Angantyr, A., Andersson, J., Aidanpaa, J.-O.: Constrained optimization based on a multiobjective evolutionary algorithm. In: Proceedings of Congress on Evolutionary Computation, pp. 1560–1567 (2003)
10. Coello, C.A.C.: Treating objectives as constraints for single objective optimization. Engineering Optimization 32(3), 275–308 (2000)
11. Deb, K., Lele, S., Datta, R.: A hybrid evolutionary multi-objective and SQP based procedure for constrained optimization. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 36–45. Springer, Heidelberg (2007)
12. Echeverri, M.G., Lezama, J.M.L., Romero, R.: An efficient constraint handling methodology for multi-objective evolutionary algorithms. Revista Facultad de Ingenieria-Universidad de Antioquia 49, 141–150 (2009)
13. Burke, E.K., Smith, A.J.: Hybrid evolutionary techniques for the maintenance schedulingproblem. IEEE Transactions on Power Systems 15(1), 122–128 (2000)

14. Fatourechi, M., Bashashati, A., Ward, R.K., Birch, G.E.: A hybrid genetic algorithm approach for improving the performance of the LF-ASD brain computer interface. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005, vol. 5 (2005)
15. Victoire, T., Jeyakumar, A.E.: A modified hybrid EP-SQP approach for dynamic dispatch with valve-point effect. International Journal of Electrical Power & Energy Systems 27(8), 594–601 (2005)
16. Hinterding, R.: Constrained parameter optimisation: equality constraints. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 687–692. IEEE, Los Alamitos (2002)
17. Peconick, G., Wanner, E.F., Takahashi, R.H.C.: Projection-based local search operator for multiple equality constraints within genetic algorithms. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 3043–3049. IEEE, Los Alamitos (2008)
18. Lin, C.Y., Wu, W.H.: Adaptive penalty strategies in genetic search for problems with inequality and equality constraints. In: IUTAM Symposium on Evolutionary Methods in Mechanics, pp. 241–250. Springer, Heidelberg (2004)
19. Shang, W., Zhao, S., Shen, Y.: A flexible tolerance genetic algorithm for optimal problems with nonlinear equality constraints. Advanced Engineering Informatics 23(3), 253–264 (2009)
20. Deb, K., Datta, R.: A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach. In: Proceedings of the Congress on Evolutionary Computation (CEC 2010), pp. 1–8 (2010)
21. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)
22. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello Coello, C.A., Deb, K.: Problem definitions and evaluation criteria for the CEC 2006: Special session on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore (2006)
23. Zavala, A.E.M., Aguirre, A.H., Diharce, E.R.V.: Continuous Constrained Optimization with Dynamic Tolerance Using the COPSO Algorithm, pp. 1–24. Springer, Heidelberg (2009)
24. Takahama, T., Sakai, S.: Solving Difficult Constrained Optimization Problems by the ε Constrained Differential Evolution with Gradient-Based Mutation, pp. 51–72. Springer, Heidelberg (2009)
25. Brest, J.: Constrained Real-Parameter Optimization with ε Self-Adaptive Differential Evolution, pp. 73–94. Springer, Heidelberg (2009)

A New Memory Based Variable-Length Encoding Genetic Algorithm for Multiobjective Optimization

Eduardo G. Carrano[1], Lívia A. Moreira[2], and Ricardo H.C. Takahashi[3]

[1] Centro Federal de Educação Tecnológica de Minas Gerais,
Department of Computer Engineering
Av. Amazonas, 7675, Belo Horizonte, MG, 30510-000, Brazil
egcarrano@dppg.cefetmg.br

[2] Centro Federal de Educação Tecnológica de Minas Gerais,
Department of Electrical Engineering
Av. Amazonas, 7675, Belo Horizonte, MG, 30510-000, Brazil
livinhaam@hotmail.com

[3] Universidade Federal de Minas Gerais,
Department of Mathematics
Av. Antonio Carlos, 6627, Belo Horizonte, MG, 31270-901, Brazil
taka@mat.ufmg.br

Abstract. This paper presents a new memory-based variable-length encoding genetic algorithm for solving multiobjective optimization problems. The proposed method is a binary implementation of the NSGA2 and it uses a Hash Table for storing all the solutions visited during algorithm evolution. This data structure makes possible to avoid the re-visitation of solutions and it provides recovering and storage of data with low computational cost. The algorithm memory is used for building crossover, mutation and local search operators with a parameterless variable-length encoding. These operators control the neighborhood based on the density of points already visited on the region of the new solution to be evaluated. Two classical multiobjective problems are used to compare two variations of the proposed algorithm and two variations of the binary NSGA2. A statistical analysis of the results indicates that the memory-based adaptive neighborhood operators are able to provide significant improvement of the quality of the Pareto-set approximations.

1 Introduction

Problems in which the value of the function is obtained through numerical simulations (complex linear system, differential equation solving, finite element or meshless models, etc) are examples of situations in which evaluating a single point can take from some seconds up to hours. In these cases, the employment of Evolutionary Algorithms (EA's) becomes restricted, since these algorithms usually demand thousands of function evaluations for reaching reasonable solutions. A possible way of making EA's well suited for handling costly function

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 328–342, 2011.

problems could be the use of some kind of memory, which could eliminate the cost of evaluating solutions previously visited by the algorithm. In the case of continuous variable problems, the memory should also be used in order to avoid the evaluation of solutions too close to solutions evaluated previously. However, the canonical implementations of the traditional EA's do not include any kind of algorithm memory and, therefore, they cannot avoid solution reevaluation.

Tabu Search [1] is an example of a meta-heuristic which employs memory. In this method, a list stores a set of movements which should not be accomplished by the method, since they are likely to lead to previously visited solutions. Since the size of this list is limited, the algorithm does not ensure non-revisiting, but it reduces its occurrence considerably. Tabu Search, however, has been studied mainly for the case of combinatorial problems.

Taking into account the time which is usually spent for reevaluating solutions, especially for costly functions of continuous variables, some recent works have been dedicated to embed memory mechanisms inside Genetic Algorithms (GA's), such as [2,3,4]. These mechanisms avoid (or at least reduce) the reevaluation of solutions. Besides, they can be used for removing duplicate solutions in the population, what can be useful since duplication removal often increases the performance of GA's [5,6]. It is important to emphasize that the memory handling aggregates additional cost to the algorithm, due to the recovering / storing operations which are continuously performed. However, this memory handling cost can be considered irrelevant if the time spent in function evaluations is considerably higher than the time required for memory operations.

Another practice which is employed by some meta-heuristics for reducing the computational cost and achieving better solutions is the use of adaptive neighborhoods. The main task of a meta-heuristic in the beginning of its execution is to find promising regions of the search space (global search) instead of providing refinement of solutions (local search). Along the iterations, the algorithm should gradually change its central task, increasing the priority of local search in order to reach near-optimal solutions. Simulated Annealing [7] and Evolution Strategies [8] are examples of algorithms which use adaptive neighborhoods which rely on some explicit parameter adaptation. Although it is not usual in GA's, some few works have proposed operators based on adaptive neighborhoods [4,9].

This paper presents a new memory based variable-length encoding genetic algorithm for solving multiobjective continuous optimization problems. The proposed algorithm is a binary implementation of the NSGA2 [10], and it uses a Hash Table for storing all the solutions already visited during its execution. This data structure makes possible to avoid the re-visitation of solutions and it provides recovering and storage of data with low computational cost. More importantly, the data structure controls the density of function evaluations in the variable space, performing a high density sampling in promising regions (regions in which more individuals "survive"). This is obtained via a variable-length encoding. This encoding scheme proportionates adaptive neighborhood by controlling the sampling density of new solutions, based on the density of points already visited in the region of the new solution. All the adaptation in the proposed algorithm is parameterless, relying only on the attempts of the Genetic Algorithm to revisit former solutions.

This paper is structures as follows:

- Section 2 explains how memory and adaptive neighborhood can be implemented inside a binary genetic algorithm.
- Section 3 introduces the algorithm operators which have been proposed based on the memory structure included in the algorithm.
- Section 4 shows the results obtained by two variations of the proposed algorithm and two variations of a canonical implementation in two classical multiobjective problems.

2 Algorithm Memory and Adaptive Neighborhood

2.1 Algorithm Memory

The association of an EA with a memory mechanism can eliminate the reevaluation of solutions and it can also reduce the redundancy in evaluations performed on solutions which are too close. This feature can be very interesting, especially because the function evaluation process is frequently the main responsible for the algorithm computational time [4]. In addtion to the elimination of the reevaluation of solutions, the use of a solution memory can avoid the inclusion of duplicate solutions in the population. It is well known that a GA requires diversity for reducing the chance of reaching premature convergence. The elimination of duplicate solutions may improve the algorithm performance, playing an important role in diversity maintenance [3,4,5,6,11]. Although checking for duplicate solutions can be computationally expensive ($O(n^2)$), it should be noticed that, when solution memory is available, this task becomes an instance of the verification of the occurrence of such solutions in the memory list.

Some efforts have been dedicated to include memory into GA's, such as [2,3,4]. Some characteristics of these references are:

- In [2], the author proposes a fixed size cache for storing the solutions evaluated recently by the algorithm. When this cache is full, the older solutions are discarded in order to allow the storage of new individuals. This algorithm does not ensure that the algorithm does not perform re-visitation, since the cache is not able to store all visited solutions. Besides, the size of the cache cannot be arbitrarily increased, since the cost of recovering a solution increases considerably with the number of points stored (linear complexity).
- In [3], the authors use a small hash table for storing the individuals evaluated by the algorithm. If this hash table reaches its maximum size, it is replaced by a bigger one. Although hash tables are usually efficient for storing data, the performance of the proposed one decreases with the number of points stored. This occurs because the hash function adopted has not been chosen taking into account the specific nature of the data generated by GA's. Besides, the transportation of the points from a hash table to a bigger one always implies a high computational cost, and it should be avoided.

- In [4], the authors propose a binary genetic algorithm associated with a binary tree for performing mono-objective optimization. This algorithm ensures non-reevaluation of solutions, and the structure of the memory adopted makes possible to perform a binary partition of the search space. This partition is used for implementing an adaptive mutation operator.

This paper proposes a modified version of the binary NSGA2, in which a hash table is incorporated into the algorithm for including memory. The function which is used for accessing the hash has been carefully designed in order to avoid degradation of the memory performance along time. This memory makes possible to generate a crossover and a mutation operator which adapt the neighborhood according to the incidence of points in the vicinity of the solution (which is estimated by the occurrence of collisions, or reevaluation attempts). A refinement is implicitly defined in the function evaluation procedure when a reevaluation attempt occurs. The main differences between this algorithm and the one shown in [4] are: *(i)* it uses a hash table instead of binary trees (the hash can be more efficient than the binary trees if the hash function is well designed); *(ii)* it contains not only an adaptive mutation operator, but also an adaptive crossover and an implicitly defined refinement operator, and; *(iii)* it is intended for multiobjective optimization.

The hash table which is used inside the proposed algorithm if briefly described bellow.

Hash Tables: Hash tables are data structures which perform data indexing and storing using key transformations [12]. Let M be the size of the hash table (number of slots in which it is possible to store data), N be the number of points which will be stored, and $P \gg N$ be the number of all possible points which could be found. Suppose that each record which will be stored has an unique key. The slot of the hash table in which a record will be stored is defined as:

$$pos_i = \text{hash}(key_i) \tag{1}$$

in which pos_i is the slot in which the record i will be stored, $\text{hash}(\cdot)$ is the hash function which was adopted, and key_i is the unique key of record i.

In hash tables, several keys are pointed to any single position, since the idea is to store any element of a set of very high cardinality in a data structure endowed with much less memory positions than such a set cardinality. This means that collisions (i.e., attempts to store different elements in a single position) are always possible. A good hash function should be designed in such a way that the probability of collisions becomes small, considering the structure of the data that should be stored. As the occurrence of some collisions is unavoidable, they should be handled by the hash table in some way. One possible way for dealing with collisions is to store the key jointly with the data and, in case of two or more records pointing to the same slot, they are stored in the same position. When the recovery of one of these records is necessary, it is recovered by a sequential search of the keys inside this slot. The cost of recovering data inside a hash table can vary from $O(1)$ (each record is stored in a different slot) to $O(n)$ (all the records are stored in the same slot). In this case the performance of the data structure depends on the quality of the hash function: a good hash function can lead to a lookup cost very close to $O(1)$.

In this paper it has been adopted an adaptation of the Pearson hash function [13] for storing the points visited by a multiobjective genetic algorithm during its execution. The number of slots in which it is possible to store data is defined by the number of bits which are used for indexing (b_{ind}), as follows:

$$M = 2^{b_{ind}} \tag{2}$$

Therefore, the size of the hash table can be varied by changing the parameter b_{ind}: *(i)* if $b_{ind} \leftarrow b_{ind} - 1$ then $M \leftarrow \frac{M}{2}$; *(ii)* if $b_{ind} \leftarrow b_{ind} + 1$ then $M \leftarrow 2 \cdot M$.

Based on this definition, each individual is divided into binary words of length b_{ind} and these words are used for finding the corresponding slot using the Algorithm 1. In this algorithm: n is the number of decision variables; n_{bits} is the number of bits per variable; T is a vector which contains a random permutation of the sequence $\{0, \ldots, (2^{b_{ind}} - 1)\}$ and $T[p]$ is the p-th term of such a permutation; $\textsc{bin2dec}(B)$ converts a binary word B, of length b_{ind}, to its equivalent decimal; $\textsc{dec2bin}(D)$ converts a decimal value D, in range $\{0, \ldots, 2^{b_{ind}} - 1\}$, to its equivalent b_{ind}-length binary word; $\oplus$ is the exclusive-or (XOR) function; $key[a, \ldots, b]$ returns the bits which lie in positions $a, \ldots, b$ of the binary word key.

Algorithm 1. Pseudo-code for Pearson Hashing

```
function PearsonHashing($key$)
    $h[0] \leftarrow 0$
    $n_{loops} \leftarrow \left\lceil \frac{n \cdot n_{bits}}{b_{ind}} \right\rceil$
    for $i = 1, \ldots, n_{loops}$ do
        $h[i] \leftarrow T[\textsc{bin2dec}(\textsc{dec2bin}(h[i-1]) \oplus key[(i-1) \cdot b_{ind} + 1, \ldots, i \cdot b_{ind}])]$
    end for
    return $h[n_{loops}]$
end function
```

It is shown in the original reference [13] that this hash function inherits two desirable characteristics from cryptographic checksum algorithms:

- A good cryptographic checksum should ensure that small changes to the data result in large and seemingly random changes to the checksum. In Pearson hash function, similar binary strings become well separated in the Hash table.
- In a good cryptographic checksum, the effect of changing some part of the data should not be eliminated by a simple change on another part. In Pearson hash function a good separation of permutations of the same series of bits (anagrams) is ensured.

It should be noticed that GA's very often produce new solutions as small perturbations of existing ones, as a result of mutation operation, and also new solutions which have block similarity with existing ones, as a result of crossover operation. This means that the Pearson hash function is well suited for storing solutions coming from GA's.

2.2 Adaptive Neighborhood

In traditional binary GA's the neighborhood is usually defined by the Hamming distance, and the mutation and crossover operators are usually built based on such a distance. In an adaptive neighborhood, the set of points which are directly achievable by a solution b at time t (the neighborhood of b at t or $\mathcal{N}(b,t)$) can vary depending on the condition of the algorithm at that time (the relation $\mathcal{N}(b,t_1) = \mathcal{N}(b,t_2)$ is not necessarily true for $t_1 \neq t_2$). The basic idea of this paper is to build an adaptive neighborhood for binary genetic algorithms by using a variable-length encoding. Each variable i in an individual is represented by a binary word of length n_{bits}, and an additional parameter $dep_i \leq n_{bits}$ is defined, indicating the depth of such a variable. This variable defines that only the dep_i most-significant bits of the variable i are changeable by the genetic operators.

On one hand, if $\exists i \in 1,\ldots,n$ such that $dep_i < n_{bits}$ than the neighborhood of the corresponding individual becomes reduced, and consequently easier to explore. On the other hand, this limited neighborhood implies on a reduction of precision of the operations, since it is not possible to change the less-significant bits of the individual.

Obviously, it is necessary to increase the depth of the variables in order to reach the desirable precision for the algorithm. This is the key of how the memory based adaptive neighborhood proposed in this paper works. Each time the algorithm tries to visit a previously visited solution, it considers the reevaluation attempt as an indication that the region around such a solution should be exploited with an enhanced precision. Therefore, the depth of one of the variables is increased, in order to increase the cardinality of the neighborhood of that solution. A detailed description of how this operation is performed, and how new solutions are generated from previously visited ones can be seen in Section 3.1.

3 Algorithm Operators

3.1 Evaluation Operator

The algorithm memory has been used for building a new evaluation operator. When this operator is called to evaluate a solution it performs one of the following actions: (i) the operator evaluates the solution if it was not visited yet; or, (ii) the operator generates an unvisited solution which is close to the original one. In any situation, only a single function evaluation is performed inside this operator and, consequently, the number of function evaluations spent by the algorithm in each generation is not increased. A scheme which describes how this operator works is given in Algorithm 2. In this algorithm: RECOVER(b)

returns *true* if the individual b was already visited by the algorithm or $false$ elsewhere; EVALUATE(b) returns the function value vector corresponding to solution b; STORE(b, f_{vl}) stores the solution b and the function vector f_{vl} on the hash table; RANDOM($\mathcal{S}$) returns an element of the set $\mathcal{S}$ at random; RandInd returns a random feasible individual; *in_dep* is the initial depth of the variables.

Algorithm 2. Pseudo-code for Evaluation Operator

```
function EvaluationOperator($b$,$dep_b$)
    $evl \leftarrow false$
    while $evl = false$ do
        $visited \leftarrow$ RECOVER($b$)
        if $visited = false$ then
            $f_{vl} \leftarrow$ EVALUATE($b$)
            STORE($b, f_{vl}$)
            $evl \leftarrow true$
        else
            $\mathcal{S}_{var} \leftarrow \emptyset$
            for $i = 1, \ldots, n$ do
                if $dep_b(i) < n_{bits}$ then
                    $\mathcal{S}_{var} \leftarrow \mathcal{S}_{var} \bigcup i$
                end if
            end for
            if $\mathcal{S}_{var} \neq \emptyset$ then
                $v \leftarrow$ RANDOM($\mathcal{S}_{var}$)
                $dep_b(v) \leftarrow dep_b(v) + 1$
                $b(dep_b(v)) \leftarrow \overline{b(dep_b(v))}$
            else
                $b \leftarrow$ RandInd
                for $i = 1, \ldots, n$ do
                    $dep_b(i) \leftarrow in\_dep$
                end for
            end if
        end if
    end while
    return $f_{vl}, b, dep_b$
end function
```

When the input parameter is an unvisited solution, the operator only performs its evaluation. Otherwise, if the solution was already visited, the operator increases the depth of one of the variables and it changes its least significant bit accessible (the bit corresponding to the new depth). In the limit case in which it is not possible to increase the depth ($dep(i) = n_{bits}$, $\forall i \in \{1, \ldots, n\}$), then a new random solution is generated to replace b. It should be noticed that this operator provides a kind of automatic refinement procedure, since the precision of the solutions which are achievable is enhanced each time the algorithm tries to reevaluate a solution.

3.2 Crossover Operator

The crossover operator which is employed inside the algorithm is inspired on the single-point per-variable crossover. A scheme of this operator can be found in Algorithm 3. In this algorithm: $c(v; \{a, \ldots, b\})$, with $a < b$, is the interval of bits $\{a, \ldots, b\}$ of variable v.

Algorithm 3. Pseudo-code for Crossover Operator

```
function CrossoverOperator($a$,$dep_a$,$b$,$dep_b$)
    $c \leftarrow a$
    $d \leftarrow b$
    for $i = 1, \ldots, n$ do
        $dep_c(i) \leftarrow \min(dep_a(i), dep_b(i))$
        $p_c \leftarrow \mathrm{RANDOM}(\{1, \ldots, dep_c(i)\})$
        $c(i; \{1, \ldots, p_c\}) \leftarrow b(i; \{1, \ldots, p_c\})$
        $dep_d(i) \leftarrow \max(dep_a(i), dep_b(i))$
        $p_d \leftarrow \mathrm{RANDOM}(\{1, \ldots, dep_d(i)\})$
        $d(i; \{1, \ldots, p_d\}) \leftarrow a(i; \{1, \ldots, p_d\})$
    end for
    return $c, dep_c, d, dep_d$
end function
```

From this algorithm, it is possible to note that the crossover point for solution c is calculated based on the minimum depth between a and b for each variable. Symmetrically, the crossover points for d are based on the maximum values between c and d.

This operator becomes exactly the single-point per-variable crossover if both depths are equal to the number of bits n_{bits}. Since the depths are usually lower than n_{bits}, the set of solutions which are reachable with this operator is reduced, and therefore, it becomes easier to explore. However, it is important to realize that this reduced set incurs in lower precision for the operator outcome.

3.3 Mutation Operator

The mutation operator which is proposed here is based on the usual flip mutation, commonly employed in binary GAs. The scheme of this operator is shown in Algorithm 4.

Algorithm 4. Pseudo-code for Mutation Operator

```
function MutationOperator($b$,$dep_b$)
    $a \leftarrow b$
    $dep_a \leftarrow dep_b$
    $v_a \leftarrow \mathrm{RANDOM}(\{1, \ldots, n\})$
    $p_a \leftarrow \mathrm{RANDOM}(\{1, \ldots, dep_a(i)\})$
    $a(v_a; p_a) \leftarrow \overline{a(v_a; p_a)}$
    return $a, dep_a$
end function
```

The only difference between this operator and the original flip mutation is that it is not possible to reach all bits of some variable i if $dep(i) < n_{bits}$. Such as for crossover, the cardinality of the neighborhood provided by this operator is controlled by the current depth of the variables.

3.4 The MBVL-NSGA2

The proposed algorithm, which is referred here as Memory-Based Variable-Length Encoding Non-Dominated Sorting Genetic Algorithm 2 (or simply MBVL-NSGA2), follows the same structure of the NSGA2 algorithm [10]. The selection is performed inside the algorithm using fast non-dominated sorting, crowding distance assignment and stochastic tournament. Each evaluated individual is stored in hash table jointly with its corresponding function and constraint values. This memory is used to avoid re-visitation and it makes possible to implement the variable length encoding, which is essential for the adaptive neighborhood operators. The variation of solutions is provided by the proposed crossover and mutation operators, as described in Sections 3.2 and 3.3 respectively.

The key feature of this algorithm is to perform the variation operations using low-cardinality neighborhoods in most part of the execution time. This makes the convergence to the vicinity of efficient solutions easier, since the cardinality of the search space becomes reduced during most part of the evolution. Such a neighborhood is adaptively changed based on the occurrence of reevaluation attempts.

3.5 Local Search Operator

Additionally to the MBAN-NSGA2, it is also proposed a Local Search Operator which inherits all the information acquired by the algorithm during function evaluations. A basic scheme of this operator is given in Algorithm 5. In this algorithm: $\mathcal{A}$ is the archive (the output of the GA); a_i is the i-th individual of set $\mathcal{A}$; $\textsc{sample}(\mathcal{A}, dep^{\mathcal{A}}, x\%)$ returns the $x\%$ individuals of $\mathcal{A}$ with higher crowding value associated; $\textsc{efficient}(\mathcal{A})$ returns the solutions of $\mathcal{A}$ which are efficient with regard to such a set.

In this operator a small part of the archive is sampled ($x\%$ solutions with higher crowding) and these solutions are successively submitted to mutation (using the operator of Section 3.3) and evaluation (using the operator of Section 3.1). This process is repeated until a maximum number of function evaluations is reached.

The goal of this operator is to provide successive improvement of the archive, using all the knowledge acquired by the algorithm during the evolutive process.

Algorithm 5. Pseudo-code for Local Search Operator

```
function MutationOperator($\mathcal{A}$)
    for $i = 1, \ldots, |\mathcal{A}|$ do
        for $j = 1, \ldots, n$ do
            $dep_i^{\mathcal{A}}(j) \leftarrow in\_dep$
        end for
    end for
    $n_{evl} \leftarrow 0$
    while $n_{evl} < max_{evl}$ do
        $\left[\mathcal{B}, dep^{\mathcal{B}}\right] \leftarrow \textsc{sample}(\mathcal{A}, dep^{\mathcal{A}}, x\%)$
        for $i = 1, \ldots, n$ do
            $\left[b_i, dep_i^{\mathcal{B}}\right] \leftarrow \textsc{MutationOperator}\left(b_i, dep_i^{\mathcal{B}}\right)$
            $\left[f_i^{\mathcal{B}}, b_i, dep_i^{\mathcal{B}}\right] \leftarrow \textsc{EvaluationOperator}\left(b_i, dep_i^{\mathcal{B}}\right)$
            $n_{evl} \leftarrow n_{evl} + 1$
        end for
        $\mathcal{A} \leftarrow \textsc{efficient}(\mathcal{A} \bigcup \mathcal{B})$
    end while
    return $\mathcal{A}$
end function
```

4 Numerical Results

Two versions of the proposed algorithm have been tested:

- MBVL-NSGA2: the original MBVL-NSGA2, with the variable-length encoding and the adaptive versions of the single point crossover and flip mutation.
- MBVL-NSGA2+LS: the same MBVL-NSGA2 of the previous item with the local search operator executed *a posteriori*. In this case the memory provided by the algorithm is used by the local search operator.

Besides, two implementations of the canonical NSGA2 have been included in the comparison:

- NSGA2: canonical binary implementation of NSGA2, with original single-point crossover and flip-mutation.
- NSGA2+LS: the same NSGA2 of the previous item with the local search operator executed a posteriori. In this case the memory is built only inside the local search operator, since there is no memory delivered by the canonical NSGA2.

The main purpose of this comparison is to evaluate the effect of the memory / variable-length based operators in the algorithm convergence (MBVL-NSGA2 *vs* NSGA2 and MBVL-NSGA2+LS *vs* NSGA2+LS) and to test if the local search operator has a positive effect in the convergence of the algorithm (MBVL-NSGA2 *vs* MBVL-NSGA2+LS and NSGA2 *vs* NSGA2+LS). The algorithms have been set with the following parameters:

- Number of runs: 30 runs (per algorithm)
- Bits for indexing hash (b_{ind}): 16 bits (2^{16} = 65,536 slots)
- Population size: 100 individuals
- Bits per variable: 16 bits
- Maximum #function evals in GA:
  - MBVL-NSGA2 and NSGA2 : 30,000
  - MBVL-NSGA2+LS and NSGA2+LS : 25,000
- Maximum #function evals in LS:
  - MBVL-NSGA2 and NSGA2 : 0
  - MBVL-NSGA2+LS and NSGA2+LS : 5,000
- Maximum archive size: 50 individuals
- Initial depth:
  - MBVL-NSGA2 and MBVL-NSGA2+LS : 8 bits
  - NSGA2 and NSGA2+LS : 16 bits
- Crossover probability: 0.80 per pair
- Mutation probability: 0.03 per bit

Notice that the algorithms spend the same number of function evaluations (30,000 evaluations), either only within the GA or divided between the GA (25,000) and the LS (5,000). The control of the archive size is performed based on the value of the crowding distance: if the size of the archive is greater than 50, then the 50 individuals with highest crowding distance values are chosen.

4.1 Comparison Methodology

The algorithm comparison which is performed here is inspired on the evaluation schemes proposed in [14,15]. This procedure can be summarized as follows:

- Each algorithm is executed k times, for a fixed number of function evaluations (stop criterion).
- For each algorithm i:
  - For each run j:
    - Evaluate the hyper-volume indicator [16], $h_{vol}(i,j)$, of the final archive achieved in the run j of the algorithm i.
  - Perform bootstrapping [17] with the values $h_{vol}(i,:)$. Each bootstrapping iteration delivers a sample of the mean of the hyper-volume associated with the algorithm i. The whole set provided by the bootstrapping is an empirical approximation of the probability distribution function (PDF) of such a mean.
- Compare the empirical PDF's of the algorithms using One-Way ANOVA [18], for a significance level $\alpha = 0.05$.

4.2 Test Problems

Kursawe Problem: The Kursawe Problem is stated as follows [19]:

$$\mathcal{X}^* = \arg\min_X \begin{cases} f_1 = \sum_{i=1}^{n-1} \left[-10 \cdot \exp\left(-0.2 \cdot \sqrt{x_i^2 - +x_{i+1}^2} \right) \right] \\ f_2 = \sum_{i=1}^{n} \left[\left| x_i^{0.8} \right| + 5 \cdot \sin\left(x_i^3 \right) \right] \end{cases} \tag{3}$$

$$\text{subject to: } -5 \leq x_i \leq 5 \quad \forall\, i = 1, \ldots, n \tag{4}$$

in which n is the number of problem variables. In the specific case of this work it has been considered $n = 10$. In this problem, it has been considered the point $y_R = [110 \;\; 9]^T$ as reference in order to evaluate the hyper-volume.

Fonseca-Flemming Problem: The Fonseca-Flemming Problem is stated as follows [20]:

$$\mathcal{X}^* = \arg\min_X \begin{cases} f_1 = 1 - \exp\left[-\sum_{i=1}^{n} \left(x_i - \frac{1}{\sqrt{n}} \right)^2 \right] \\ f_2 = 1 - \exp\left[-\sum_{i=1}^{n} \left(x_i + \frac{1}{\sqrt{n}} \right)^2 \right] \end{cases} \tag{5}$$

$$\text{subject to: } -4 \leq x_i \leq 4 \quad \forall\, i = 1, \ldots, n \tag{6}$$

in which n is the number of problem variables. In the specific case of this work it has been considered $n = 10$. The reference point in this problem has been set to $y_R = [5 \;\; 5]^T$.

4.3 Results

A summary of the results observed for the algorithm outcomes in the two test problems can be seen in Table 1. This table shows the results observed with regard to hyper-volume indicator in the format *mean / standard deviation* (higher values of hyper-volume mean better Pareto-set approximations).

Table 1. Summary of results

		Kursawe	Fonseca & Fleming
Index	Algorithm	*mean / std*	*mean / std*
$A1$	NSGA2	899.16 / 2.94	21.83 / 0.10
$A2$	MBVL-NSGA2	901.91 / 2.17	22.83 / 0.06
$A3$	NSGA2+LS	909.64 / 2.62	22.52 / 0.08
$A4$	MBVL-NSGA2+LS	917.22 / 2.44	23.10 / 0.09

In the case of Kursawe problem, after running ANOVA, it has been observed $p_{value} = 4.86 \cdot 10^{-42}$, what indicates that at least one algorithm is significantly different from the other three ones. In a multiple comparison test it was noticed that all algorithms are statistically different, and therefore it is possible to say that $A4$ is better than $A3$, which is better than $A2$, that is better than $A1$.

In the case of Fonseca & Fleming problem, the ANOVA test indicates $p_{value} = 4.27 \cdot 10^{-75}$, what means that there is a significant difference amongst the algorithms. Once again, the multiple comparison test indicates that each algorithm is statistically different from the other ones. However, in this case, the ranking obtained was slightly different: 1^{st} : $A4$, 2^{nd} : $A2$, 3^{rd} : $A3$ and 4^{th} : $A1$.

Figure 1 shows a boxplot of these results.

It is possible to note that, in the Kursawe problem, the memory based / adaptive neighborhood operators have a profitable effect on the quality of the final solution, since the algorithm $A2$ obtained better Pareto approximations

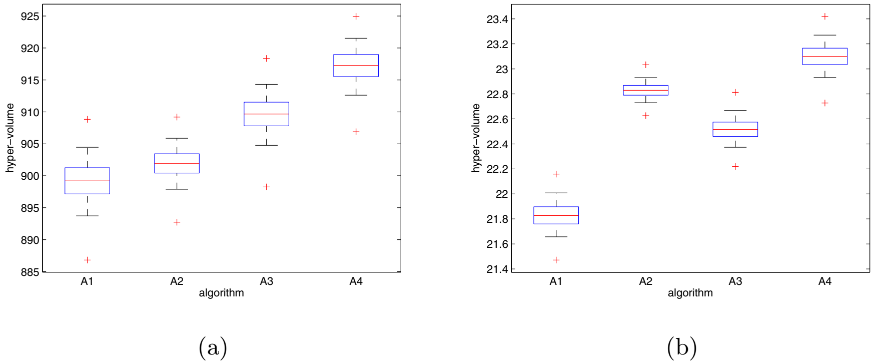


Fig. 1. Hyper-volume boxplot. Test problems: (a) Kursawe; (b) Fonseca & Fleming.

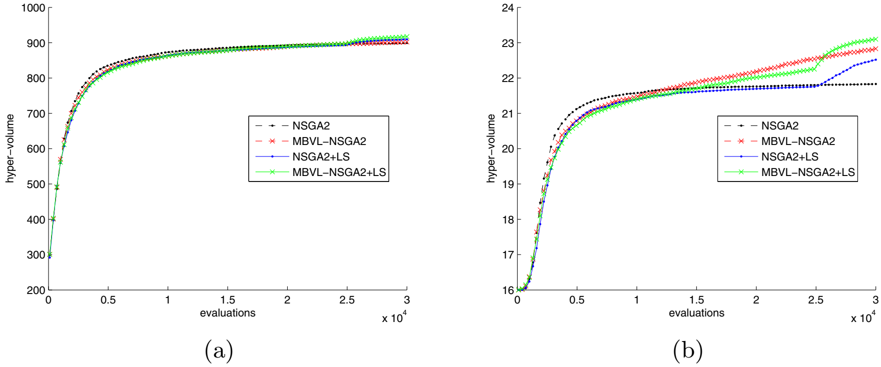


Fig. 2. Hyper-volume evolution: (a) Kursawe problem; (b) Fonseca & Fleming problems

than $A1$. The local search operator has also improved the convergence of the algorithms, since $A3$ and $A4$ have outperformed $A1$ and $A2$, respectively. The algorithm $A4$, which combines the new operators with local search, was the one which has shown better performance amongst the four tested algorithms.

The difference between the ranks achieved in the two problems can be explained by the fact that the adaptive neighborhood operators introduce a more beneficial effect in the second problem. The same analysis, which indicates that the neighborhood operators and the local search are able to improve the convergence, is also valid in the case of Fonseca & Fleming problem, since $A2$ is better than $A1$ (adaptive operators), $A3$ is better than $A1$ and $A4$ is better than $A2$ (local search).

The Figure 2(a) shows how the hyper-volume evolves during the execution of the algorithm in the Kursawe problem. From this figure it is possible to see that the algorithms have very similar behavior. Some differences between the algorithms can be noted at the end of the algorithm execution, when the local search and the memory based operators have a beneficial effect on the quality of the archive. Finally, the improvement provided by local search can be easily noted in the evolution curve of the algorithms in Fonseca & Fleming problem, shown in Figure 2(b) (between 25,000 and 30,000 function evaluations).

5 Conclusions

A binary variable-depth encoding of the solutions of a continuous-variable genetic algorithm has been used here in order to provide an adaptation mechanism that provides a more intensive exploitation of regions in which the GA performs more trials of new solutions. The adaptation is based on the information that a solution produced by the GA operations has already been evaluated. This information about all the solutions that have already been visited by the algorithm is stored in a hash table that is particularly chosen in order to efficiently store the data coming from a GA.

The multiobjective genetic algorithms constructed on the basis of such variable-lenght encoding and hash table memory have been tested in standard test problems, showing a significant improvement in solution quality, when compared with the canonical versions of the same algorithms.

Acknowledgments

The authors would like to thank the Brazilian agencies Fapemig, CAPES and CNPq for the financial support.

References

1. Glover, F.W., Laguna, M.: Tabu Search, 1st edn. Springer, Heidelberg (1998)
2. Kratica, J.: Improving performances of the genetic algorithm by caching. Computers and Artificial Intelligence 18, 271–283 (1999)

3. Povinelli, R.J., Feng, X.: Improving genetic algorithms performance by hashing fitness values. In: Proc. Artificial Neural Networks on Engineering (1999)
4. Yuen, S.Y., Chow, C.K.: A genetic algorithm that adaptively mutates and never revisits. IEEE Transactions on Evolutionary Computation 13, 454–472 (2009)
5. Mauldin, M.L.: Maintaining diversity in genetic search. In: Proc. National Conference on Artificial Intelligence (1984)
6. Friedrich, T., Hebbinghaus, N., Neumann, F.: Rigorous analyses of simple diversity mechanisms. In: Proc. Genetic and Evolutionary Computation Conference (2007)
7. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
8. Beyer, H., Schwefel, H.: Evolution strategies - A comprehensive introduction. Natural Computing 1, 3–52 (2002)
9. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3, 124–141 (1999)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
11. Ronald, S.: Duplicate genotypes in a genetic algorithm. In: Proc. IEEE International Conference on Evolutionary Computation (1998)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. The MIT Press, Cambridge (2001)
13. Pearson, P.K.: Fast hashing of variable-length text strings. Computing Practices 33, 677–680 (1990)
14. Carrano, E.G., Takahashi, R.H.C., Wanner, E.F.: An enhanced statistical approach for evolutionary algorithm comparison. In: Proc. Genetic and Evolutionary Computation Conference (2008)
15. Carrano, E.G., Wanner, E.F., Takahashi, R.H.C.: A multi-criteria statistical based comparison methodology for evaluating evolutionary algorithms. IEEE Transactions on Evolutionary Computation (to appear, 2011)
16. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph.D. dissertation, Computer Engineering and Networks Laboratory - Swiss Federal Institute of Technology Zurich (1999)
17. Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7, 1–26 (1979)
18. Lindman, H.R.: Analysis of Variance in Complex Experimental Designs. W. H. Freeman & Co, San Francisco (1974)
19. Kursawe, F.: A variant of evolution strategies for vector optimization. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 193–197. Springer, Heidelberg (1991)
20. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3, 1–16 (1995)

A Concentration-Based Artificial Immune Network for Multi-objective Optimization

Guilherme Palermo Coelho and Fernando J. Von Zuben

Laboratory of Bioinformatics and Bioinspired Computing (LBiC)
Department of Computer Engineering and Industrial Automation (DCA)
School of Electrical and Computer Engineering (FEEC)
University of Campinas (Unicamp)
Campinas, SP, Brazil
{gcoelho,vonzuben}@dca.fee.unicamp.br

Abstract. Until recently, the main focus of researchers that develop algorithms for evolutionary multi-objective optimization has been the creation of mechanisms capable of obtaining sets of solutions that are as close as possible to the true Pareto front of the problem and also as diverse as possible in the objective space, to properly cover such front. However, an adequate maintenance of diversity in the decision space is also important, to efficiently solve several classes of problems and even to facilitate the post-optimization decision making process. This aspect has been widely studied in evolutionary single-objective optimization, what led to the development of several diversity maintenance techniques. Among them, the recently proposed concentration-based artificial immune network (cob-aiNet), which is capable of self-regulating the population size, presented promising results in multimodal problems. So, it is extended here to deal with multi-objective problems that require a proper maintenance of diversity in the decision space.

Keywords: artificial immune systems, multi-objective optimization, diversity in decision space, immune networks.

1 Introduction

Since the early times in the field of Evolutionary Multi-Objective Optimization (EMO) [6], the main goals pursued by the great majority of problem-solving techniques have been to find a set of solutions that better approximates the *Pareto front* of a given problem and also presents high diversity in the objective space, so that the front could be properly covered. These two fundamental goals have driven most EMO researchers to basically focus their work on mechanisms that act in the *objective space* of the problems. However, a proper attention to the behavior of the nondominated solutions in the decision space, also known as the *Pareto set* of a given problem, is also very important. The correct understanding of how these solutions are distributed in the *decision space* of a problem and the proposal of mechanisms that stimulate diversity in both the objective

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 343–357, 2011.

and the decision spaces may be mandatory for efficiently solving several classes of problems, mainly those with multimodal characteristics. Besides, in several real-world applications, particularly when the preferences and constraints of the decision maker are not well-known a priori, a good approximation of both the Pareto front and the Pareto set tends to facilitate the decision making process [23].

One of the pioneering ideas to explore decision space diversity into multi-objective optimization problems was introduced in the early 90s with the first NSGA proposal [19], which included fitness sharing among its mechanisms. However, after that, the EMO research focus returned to objective space diversity for almost one decade until 2003, when the GDEA [20] proposal considered *diversity in the decision space* as an extra objective for the search. Since then, besides some studies about the diversity issue itself and the proposal of new metrics that consider diversity [16,17,21], a few algorithms were introduced. In 2005, two techniques were proposed: the first one was the omni-optimizer [8], which is an algorithm that extends the idea of the NSGA-II [7] to consider diversity in the decision space, and is capable of solving both single and multi-objective optimization problems; and the second one was Chan & Ray's algorithm [4], which is an evolutionary algorithm that evaluates the quality and diversity of each solution in the population (both in objective and decision spaces) according to two metrics, the Lebesgue Contribution and Neighborhood Count. Two algorithms were also proposed in 2009: in [18], Shir *et al.* included diversity of the decision space into the CMA-ES niching framework; and in [23] Zhou *et al.* proposed a clustering-based EDA that stimulates diversity in the decision space by means of its reproduction operator.

On the other hand, in single-objective optimization, the behavior of candidate solutions in the decision space has been well explored in the literature for a long time, and several different techniques capable of stimulating and maintaining diversity have been proposed. Among them, the *Artificial Immune Systems* (AIS) [3] framework is particularly interesting, given that some of the algorithms proposed according to this framework present intrinsic mechanisms to maintain diversity in the decision space [11].

The general idea behind *Artificial Immune Systems* (AIS) [3] is to try to explore and reproduce concepts and mechanisms of the natural immune system of vertebrates in computer science. Following this idea, two distinct lines of research have emerged in the AIS field: one that searches for a better understanding of the natural immune system by developing computer-aided models; and one that tries to develop immune-inspired algorithms capable of dealing with a wide range of practical problems, such as in data analysis and optimization processes. Searching for a way to approximate these two lines of research, Coelho & Von Zuben [5] recently proposed an immune-inspired algorithm intended to explore one of the computational models of the immune system to solve single-objective optimization problems, what led to the *Concentration-based Artificial Immune Network*, or just *cob-aiNet*.

In [5], the diversity maintenance mechanisms of cob-aiNet were extensively compared to the mechanisms of two other popular immune inspired algorithms, and its results were consistently superior. Besides, cob-aiNet was also compared to several state-of-the-art algorithms from the literature, and the obtained results showed that it is a competitive approach. Therefore, the main goal of this paper is to extend cob-aiNet to solve multi-objective optimization problems, more specifically multimodal multi-objective problems, so that cob-aiNet's diversity mechanisms could be properly extended to deal not only with diversity in the objective space (as several immune-inspired algorithms for multi-objective optimization do) but also in the decision space.

This paper is organized as follows: in Section 2 the new concentration-based immune network for multi-objective optimization (*cob-aiNet*[MO]) will be described in details; in Section 3 the benchmark problems adopted in this work to evaluate the performance of cob-aiNet[MO] will be presented, together with the experimental methodology employed, the obtained results and the discussion of such results; and, finally, in Section 4 some final comments and future perspectives for this research will be given.

2 The Concentration-Based Immune Network for Multi-objective Optimization

The *Concentration-based Immune Network for Multi-objective Optimization* (cob-aiNet[MO]) is an extension of the original *Concentration-based Immune Network for Continuous Optimization* (cob-aiNet). As the two immune theories taken as inspiration for cob-aiNet are also the basis of cob-aiNet[MO], both algorithms have mechanisms that try to mimic the immune behavior proposed by the *Clonal Selection* [2] and *Immune Network* [14] theories.

According to the Clonal Selection principle [2], when the organism is invaded by a foreign pathogen, the molecular signatures of this pathogen (its *antigens*) are identified and certain immune cells begin to suffer a cloning process and proliferate. During this proliferation, these cells also suffer a controlled mutation combined with a selective pressure, so that only those cells capable of producing antibodies[1] with the best affinity with the antigens are selected to remain in the population. Therefore, the Clonal Selection principle states that the components of the immune system remain in standby until foreign antigenic stimuli trigger the immune response.

On the other hand, the Immune Network theory [14] proposes that the cells of the immune system are not only capable of recognizing antigens but also each other, which may cause two different behaviors: a positive response, characterized by the activation of the recognizing cell followed by cloning, mutation and secretion of antibodies; and a negative response, characterized by tolerance and even the suppression of the recognized cell. Therefore, the Immune Network theory states that the main aspects of the natural immune system are emergent properties, consequence of regulatory mechanisms intended to maintain the network in a given *dynamic steady state* [1].

[1] Antibodies are molecular structures that bind to the antigens and signal that they must be eliminated.

Algorithm 1. Main steps of cob-aiNet[MO].

```
Parameters:
- $nAB$: initial number of antibodies;
- $maxAB$: maximum number of antibodies;
- $nC^{max}$: maximum number of clones per antibody;
- $nC^{min}$: minimum number of clones per antibody;
- $C_0$: initial concentration;
- $\sigma_s$: suppression threshold;
- $\beta^i/\beta^f$: initial/final mutation parameter;

1- Randomly create the initial population;
while stop criterion not satisfied do
    2- Clone individuals in the population;
    3- Apply hypermutation to the clones;
    4- Select cells for the next generation (with insertion);
    5- Update concentration;
    8- Eliminate cells with null concentration;
end while
9- Eliminate dominated cells from the population;
```

Although the Immune Network principles have been very popular in the AIS literature, its computational implementations are generally restricted to the negative response of immune cells. Besides, such behavior is usually implemented in the form of eventual comparisons (during runtime) among all the cells in the population and the elimination of the worst subset of cells (those that present lower affinity with the antigens) from those that are closer than a given threshold to each other [11]. On the other hand, cob-aiNet introduced a distinct approach, with the implementation of the Immune Network mechanisms partially based on the 2D shape-space immune network model of Hart *et al.* [13].

In this section, the overall structure of the cob-aiNet[MO] algorithm will be presented, with emphasis on mechanisms and operators devoted to the multi-objective scenario. The readers are referred to [5] for a full description of cob-aiNet, including a local search operator that has been discarded here.

The main steps of cob-aiNet[MO] are given in Alg. 1. As we can see, the overall structure of this algorithm is similar to other immune network-inspired algorithms from the literature [11], except for the inclusion of the concentration update steps, the novel concentration-based suppression mechanism, and the selection operator that also performs insertion. However, there are also some differences in the remaining steps, that will be highlighted in what follows.

2.1 Representation and Affinity Metrics

Like the original cob-aiNet, the individuals in cob-aiNet[MO]'s population (or cells/antibodies, using AIS terminology) correspond to candidate solutions for the problem and are encoded as real-valued vectors.

The basis of immune-inspired algorithms is the *affinity metrics*, evaluated among each antibody and the antigens of the model (generally associated with the problem), and also among all the antibodies in the population. As both cob-aiNet and cob-aiNet[MO] deal with optimization problems, there are no explicit antigens in this case, so the affinity among antibodies and antigens are considered to be a *fitness-like* function [3] that evaluates the quality of each cell in the population.

In cob-aiNet, such *affinity with antigens* metric (or just *fitness*) was simply taken as the value of the objective function being optimized, normalized in the interval $[0,1]$. As the same approach is not possible in the multi-objective scenario (there are multiple functions to be optimized simultaneously) and as this affinity metric is very important to several mechanisms of the algorithm (e.g. the fitness-proportional hypermutation operator [5]), we have decided to adopt a *fitness-assignment* strategy in cob-aiNet[MO] instead of a direct *dominance-based ranking* of the solutions. In this way, the remaining operators of the algorithm require a minimum amount of modification.

Therefore, in this work we adopted the fitness-assignment proposed in the SPEA2 algorithm [24], normalized in the interval $[0,1]$ and inverted, as the fitness function f_{SPEA2} of the SPEA2 algorithm was originally proposed to be minimized. Therefore, in cob-aiNet[MO], the affinity $f_i^{Ag}(t)$ of each antibody i of the population with the antigens at iteration t is given by $f_i^{Ag}(t) = 1 - \underline{f}^i_{SPEA2}$, where $\underline{f}^i_{SPEA2}(t)$ is the normalized SPEA2 fitness of solution i.

Another reason for the adoption of the SPEA2 fitness is that it combines, into a single metric, aspects of dominance relations among the solutions in the population and also of the spread of individuals in the *objective space*. Therefore, by using this metric in cob-aiNet[MO], the evolutionary mechanisms of the algorithm will be directly responsible for finding a set of solutions as close as possible to the Pareto front of the problem and also as spread as possible in the objective space, thus allowing the diversity mechanisms of the algorithm to focus solely on the decision space.

The second affinity metric of the cob-aiNet[MO] algorithm evaluates the degree of similarity between a given cell i in the population and all the other individuals that are closer to i than a given threshold σ_s. In cob-aiNet[MO], the same metric proposed in cob-aiNet was adopted, and it is given in Eq. 1. This affinity among antibodies is essential to define a decrease in the concentration of a given cell, as will be seen in Sect. 2.2.

$$f_i^{Ab}(t) = \begin{cases} \dfrac{\sum_{j \in J} C_t^j \cdot [\sigma_s - d(i,j)]}{\sum_{j \in J} C_t^j} & \text{if } J \neq \emptyset \\ 0 & \text{otherwise} \end{cases}, \qquad (1)$$

where $f_i^{Ab}(t)$ is the total affinity among antibody i and the remaining antibodies at iteration t, J is the set of antibodies better than i and within a radius σ_s (defined by the user) from i, C_t^k is the concentration of antibody k at iteration t and $d(k,l)$ is the Euclidean distance between antibodies k and l.

From Eq. 1, it is possible to notice that only antibodies closer to i than a given threshold and also better than i ($f_j^{Ag}(t) > f_i^{Ag}(t)$) are considered in the calculation of the total affinity $f_i^{Ab}(t)$, and their contribution is proportional to how close they are from i, weighted by their concentration. As it will be seen in next section, the affinity $f_i^{Ab}(t)$ among antibodies may lead to a decrease in the concentration of antibody i when i is in the vicinity of better cells, and this decrease is proportional to the concentration of such cells.

2.2 Concentration Model and Suppression

Both in cob-aiNet and cob-aiNet[MO], the dynamics of the algorithm is controlled by the *concentration* C_t^i of each cell i, which is directly related to the fitness of the individual and the presence of such individual in regions crowded by better solutions ($f_i^{Ab}(t) > 0$). This concentration not only influences the affinity among antibodies, but it is also responsible for the determination of the number of clones that will be generated for each cell and when a given antibody is going to be eliminated from the population (when its concentration becomes null).

The concentration C_t^i of antibody i starts with value C_0 (defined by the user) and it is updated at the end of each iteration according to Eq. 2:

$$C_{t+1}^i = \alpha C_t^i - f_i^{Ab}(t), \tag{2}$$

where C_{t+1}^i and $C_t^i \in [0,1]$ are, respectively, the new and old concentration of antibody i, $f_i^{Ab}(t)$ is the affinity of antibody i with the other cells of the population and α is given by Eq. 3:

$$\alpha = \begin{cases} 1 + 0.1 f_i^{Ag}(t) & \text{if } f_i^{Ab}(t) = 0 \\ 0.85 & \text{otherwise} \end{cases}, \tag{3}$$

where $f_i^{Ag}(t)$ is the fitness of antibody i.

Therefore, in cob-aiNet[MO], the best solutions in their neighborhood (those with $f^{Ab}(t) = 0$) suffer an increase of up to 10% in concentration, proportionally to their fitness (which is normalized in $[0,1]$ – see Sect. 2.1), while individuals within the region of influence of better solutions suffer a decrease of concentration of at least 15%. These values of 10% and 15% were empirically defined.

2.3 Cloning and Hypermutation

In both cob-aiNet and cob-aiNet[MO], the evolution of candidate solutions in the population occurs by means of a combination of two mechanisms: *cloning* and *hypermutation*. In the cloning phase, at each iteration t a given number of clones nC_t^i is generated for each antibody i in the population, and nC_t^i is determined according to the concentration of each cell C_t^i: nC_t^i varies linearly with the concentration C_t^i, in the interval between nC^{min} and nC^{max}.

After the cloning phase, the new generated clones suffer a process of hypermutation with genetic variability inversely proportional to their fitness (better individuals suffer smaller variations and vice-versa), according to Eq. 4:

$$Ab_{t+1}^i = Ab_t^i + \beta_t \cdot e^{-f_i^{Ag}(t)} \cdot \mathcal{N}(0,1), \tag{4}$$

where Ab_{t+1}^i is the new antibody i, Ab_t^i is the original antibody i, $f_i^{Ag}(t)$ is the fitness of antibody i and $\mathcal{N}(0,1)$ is a random value of Gaussian distribution with mean 0 and variance 1. The parameter β_t dynamically varies during the execution of the algorithm, and it is responsible for a gradual modification of the behavior of the search, from a highly exploratory stage at the initial iterations to a stage of fine-tuning of the solutions at the final iterations. For further information about the hypermutation mechanism of cob-aiNet[MO], the reader is referred to cob-aiNet's original paper [5].

2.4 Selection and Insertion of Individuals

After cloning and hypermutation, a selection mechanism is applied to each pool of cells (original antibody and its mutated clones), to select which solutions will remain in the population in the next iteration. As in cob-aiNet, this selection operator is also responsible for the insertion of new individuals in the population, differently from other artificial immune networks in the literature, which generally insert a predefined number of new randomly generated antibodies [11,12,22].

The selection and insertion mechanism basically consists of three steps. First, the best cell in its pool (the one with the best fitness $f_i^{Ag}(t)$) is selected as the representative of the pool in the next iteration of the algorithm, and inherits the concentration of the original cell. Then, all *globally non-dominated cells* from all the pools, that haven't been selected in the previous step, are selected as candidate cells to be inserted into the population as new cells. And, finally, all these candidate cells are compared to the antibodies already in the population and, if a given cell is not in the neighborhood of any other cell (not closer than σ_s), it is inserted into the population as a new antibody, with initial concentration C_0.

The insertion of new individuals in cob-aiNet[MO] is intrinsically associated with the dynamic variation of β_t (see Sect. 2.3), as insertions occur especially during the exploration phase of the algorithm when the genetic variation of the clones is higher. After these initial iterations, the algorithm gradually starts to shift its behavior to the fine-tuning stage, and the number of clones mutated to regions of the decision space far from the original antibody also gradually decreases.

One last aspect that is important to highlight is that these insertion and suppression mechanisms allow cob-aiNet[MO] to dynamically adjust its population size during runtime, which is a common aspect in artificial immune networks.

3 Experimental Results

In this section, the benchmark problems and the methodology adopted in the experimental analysis will be presented, together with a thorough discussion of the obtained results.

3.1 Benchmark Problems

In this introductory paper of the cob-aiNet[MO] algorithm, the proposed technique was applied to four multimodal multi-objective benchmark problems that present distinct characteristics, so that the algorithm's capabilities to find and maintain good and diverse solutions (both in objective and decision spaces) could be evaluated.

The first problem adopted in this work was the bi-objective multi-global problem proposed by Deb & Tiwari [8] to evaluate their omni-optimizer algorithm, which is given by Eq. 5:

$$\text{Minimize} \begin{cases} f_1(\boldsymbol{x}) = \sum_{i=1}^{n} \sin(\pi \cdot x_i) \\ f_2(\boldsymbol{x}) = \sum_{i=1}^{n} \cos(\pi \cdot x_i) \end{cases}, \tag{5}$$

where $n = 5$ and $x_i \in [0, 6]\ \forall i \in \{1, \cdots, n\}$.

The second benchmark problem was one of the functions from the EBN class of problems, proposed by Emmerich *et al.* [9]. The EBN problems are interesting because all the solutions in the domain $[0, 1]^n$ are Pareto-optimal, what makes these problems useful to evaluate the capabilities of a given algorithm to spread the solutions in the decision space. In this work, we considered the EBN problem with linear Pareto front, given by Eq. 6:

$$\text{Minimize} \begin{cases} f_1(\boldsymbol{x}) = \frac{1}{n} \cdot \sum_{i=1}^{n} |x_i| \\ f_2(\boldsymbol{x}) = \frac{1}{n} \cdot \sum_{i=1}^{n} |x_i - 1| \end{cases}, \tag{6}$$

where $n = 10$ and $x_i \in [0, 1]\ \forall i \in \{1, \cdots, n\}$.

The third problem, named Two-on-One, was proposed by Preuss *et al.* [16] and consists in a bidimensional bi-objective function, in which the first criterion is a 4th-degree polynomial and the second one is the sphere function. From the five variations of this problem presented in [16], we adopted here the fourth one (*Two-on-One 4*), defined in Eq. 7:

$$\text{Minimize} \begin{cases} f_1(x_1, x_2) = x_1^4 + x_2^4 - x_1^2 + x_2^2 - 10x_1x_2 + 0.25x_1 + 20 \\ f_2(x_1, x_2) = (x_1 - 1)^2 + (x_2)^2 \end{cases}, \tag{7}$$

where $x_i \in [-3, 3]\ \forall i \in \{1, 2\}$.

Finally, the last benchmark problem adopted in this work (given by Eq. 8) was one of the instances of the family of problems introduced by Emmerich & Deutz [10], in which the Pareto fronts present spherical or super-spherical geometry. In this problem (named here *Lamé Supersphere*), the Pareto set consists in sets of solutions placed on equidistant parallel line segments.

$$\text{Minimize} \begin{cases} f_1(\boldsymbol{x}) = (1+r)\cdot\cos(x_1) \\ f_2(\boldsymbol{x}) = (1+r)\cdot\sin(x_1) \end{cases}, \tag{8}$$

where $r = \sin^2(\pi \cdot \xi)$, $\xi = \frac{1}{n-1}\sum_{i=2}^{n} x_i$, $n = 4$, $x_1 \in [0, \frac{\pi}{2}]$, and $x_i \in [1, 5]$ $\forall i \in \{2, \cdots, n\}$.

3.2 Methodology

In order to verify how cob-aiNet[MO] performs when compared to other algorithms from the literature, we have chosen four distinct contenders: (*i*) the popular *nondominated sorting genetic algorithm II* (NSGA-II), proposed by Deb *et al.* [7], which is one of the most popular evolutionary algorithms for multi-objective optimization; (*ii*) the *omni-optimizer*, proposed by Deb & Tiwari [8]; (*iii*) the *Vector Immune System* (VIS) proposed by Freschi & Repetto [12], which is, as cob-aiNet[MO], an immune network-based algorithm; and (*iv*) the proposal of Chan & Ray [4] (in this section, we will refer to this algorithm as *KP1*).

Each of the five algorithms was applied to all the four benchmark problems previously presented and evaluated according to three metrics: (*i*) *hypervolume* [6], which corresponds to the volume covered by the non-dominated solutions found by each algorithm, bounded by a given reference point; (*ii*) *spacing* [6], which describes the spread in the objective space of the non-dominated solutions found by each algorithm; and (*iii*) the hypercube-based diversity metric proposed in [7], to evaluate the spread of solutions in the decision space. According to this diversity metric, the decision space (domain of the problem) is divided into a predefined number of hypercubes and the number of hypercubes occupied by solutions is counted. In this work, we divided each dimension of each problem into 20 intervals, to obtain the hypercubes.

The average and standard deviation of these three metrics were calculated for each algorithm for each of the benchmark problems after 30 repetitions, and the pairwise significance among the results of cob-aiNet[MO] and all the other four algorithms was evaluated with the *Wilcoxon's Rank Sum test* [15] (with significance threshold equal to 0.05). In each repetition, the algorithms were allowed to evolve until $4 \cdot 10^4$ fitness evaluations were achieved.

The parameters of each algorithm were empirically adjusted for all the problems during a series of preliminary experiments, in order to promote competitive performances for all of them.

3.3 Results and Discussion

For all the problems studied here, cob-aiNet[MO] was run with the following parameters: initial population size $nAB = 100$; minimum and maximum number of clones $nC^{min} = 3$ and $nC^{max} = 7$, respectively; initial concentration $C_0 = 0.5$; maximum population size $maxAB = 250$ cells for problems *Deb & Tiwari* and *Two-on-One 4*, and $maxAB = 150$ cells for problems *EBN* and *Lamé Supersphere*; $\beta^i = 2.50$ and $\beta^f = 0.001$ for *Lamé Supersphere* and $\beta^i = 1.00$

and $\beta^f = 0.001$ for the other problems; and, finally, neighborhood threshold $\sigma_s = 0.060$, 0.004, 0.150 and 0.070 for problems *Deb & Tiwari*, *Two-on-One 4*, *EBN* and *Lamé Supersphere*, respectively.

The parameters of NSGA-II and omni-optimizer were the same for all the problems: population size equal to 100; mutation probability equal to $1/n$, where n is the dimension of the problem; crossover probability equal to 1.0; and crossover and mutation distribution index equal to 20. For the omni-optimizer, the parameter δ for the ϵ-dominance concept was set to $\delta = 0.001$.

The KP1 algorithm was evaluated with: population size equal to 100; mutation probability equal to $1/n$, where n is the dimension of the problem; crossover probability equal to 0.80; and crossover and mutation distribution index also equal to 20.

Finally, VIS was evaluated with the following parameters: initial population size equal to 200; number of clones per cell equal to 5; percentage of random cells equal to 20; 5 iterations between two suppression steps; and α equal to 2.5 for *Lamé Supersphere* and 1.0 for the other problems.

The average and standard deviation of the results obtained by each algorithm for each problem are given in Tab. 1. Those results in which the null hypothesis[2] can be rejected with significance 0.05 are marked with "*". To calculate the hypervolume for each solution set, the reference points for problems *Deb & Tiwari*, *EBN*, *Two-on-One 4* and *Lamé Supersphere* were $[0, 0]$, $[1, 1]$, $[21, 10]$ and $[1, 1]$, respectively.

Considering the hypervolume metric, it is possible to notice from Tab. 1 that cob-aiNet[MO] presented the best average hypervolume for problem *Two-on-One 4* (although it is not possible to reject the null hypothesis with significance 0.05 when comparing cob-aiNet[MO] and omni-optimizer), while the best results for problems *Deb & Tiwari*, *EBN* and *Lamé Supersphere* were obtained by KP1.

Although cob-aiNet[MO] presented the best results w.r.t. the hypervolume only for problem *Two-on-One 4*, it is possible to see from Fig. 1 (graphical representation – in the objective space – of the final set of non-dominated solutions) that cob-aiNet[MO] obtained good results for all the problems, in opposition to the other techniques that presented degraded performances in some of the problems (e.g. omni-optimizer in problem *EBN* and VIS in problems *EBN* and *Two-on-One 4*). The results shown in Fig. 1 correspond to the repetition that led to the best hypervolume for each algorithm.

With respect to the spacing metric, cob-aiNet[MO] obtained the best results for the *Lamé Supersphere*, while NSGA-II was the best one in *Deb & Tiwari*'s problem (although its results were not significantly different from those of cob-aiNet[MO]). For the other two problems, omni-optimizer obtained final sets of non-dominated solutions with the best spacing. However, it is important to notice that the spacing metric only indicates how well-spaced the non-dominated solutions obtained by a given algorithm are, and not whether these solutions

[2] In the Wilcoxon Rank Sum test, the null hypothesis states that the data in two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent samples from identical continuous distributions with equal medians.

Table 1. Average ± std. deviation of the *hypervolume*, *spacing*, *hypercube-based diversity* and overall rank. The best results are marked in bold. In the first column, "D&T" corresponds to the *Deb & Tiwari* problem, "2-on-1" to the *Two-on-One 4* and "Lamé" to the *Lamé Supersphere*. Those results in which the null hypothesis (when compared to cob-aiNet[MO]) can be rejected with significance 0.05 are marked with "*".

Hypervolume (higher is better)

	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1
D&T	15.856 ± 0.816	16.497 ± 0.704*	16.279 ± 0.783	15.947 ± 0.817	**18.395 ± 0.928***
EBN	0.499 ± 0.037	0.392 ± 0.080*	0.412 ± 0.024*	0.374 ± 0.090*	**0.597 ± 0.021***
2-on-1	**112.057 ± 7.181**	89.368 ± 32.456*	99.692 ± 22.677	83.608 ± 22.703*	102.179 ± 20.869
Lamé	0.364 ± 0.043	0.399 ± 0.028*	0.400 ± 0.049*	0.357 ± 0.039	**0.419 ± 0.054***

Spacing (lower is better)

	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1
D&T	0.0358 ± 0.0139	**0.0347 ± 0.0035**	0.0470 ± 0.0069*	0.0589 ± 0.0264*	0.0592 ± 0.0068*
EBN	0.0158 ± 0.0039	0.0179 ± 0.0032*	**0.0103 ± 0.0055***	0.0149 ± 0.0092*	0.0116 ± 0.0011*
2-on-1	0.1827 ± 0.0283	0.1099 ± 0.0285*	**0.0763 ± 0.0084***	0.1979 ± 0.0828	0.0963 ± 0.0098*
Lamé	**0.0050 ± 0.0019**	0.0113 ± 0.0136*	0.0249 ± 0.0295*	0.0166 ± 0.0154*	0.0114 ± 0.0016*

Hypercube-based Diversity (higher is better)

	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1
D&T	**78.667 ± 3.698**	40.867 ± 4.876*	63.533 ± 5.782*	32.433 ± 4.297*	43.200 ± 5.774*
EBN	**134.733 ± 4.025**	65.867 ± 6.642*	99.300 ± 0.915*	29.400 ± 4.553*	92.000 ± 3.006*
2-on-1	11.600 ± 0.621	11.333 ± 0.758	10.933 ± 1.596	11.400 ± 1.453	**12.433 ± 1.357***
Lamé	**108.700 ± 4.900**	83.933 ± 4.571*	85.100 ± 3.782*	67.467 ± 7.669*	65.600 ± 3.369*

Overall Rank (from the above results – lower is better)

	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1
	2.33 ± 1.50	3.25 ± 1.14	2.67 ± 1.30	4.33 ± 0.78*	2.42 ± 1.44

present a maximum spread. This explains the victory of the omni-optimizer for problems *EBN* and *Two-on-One 4*, even though it was not able to cover the whole Pareto fronts for most of its executions for these problems, as illustrated in Fig. 1.

Now focusing on the diversity of the final sets of solutions in the decision space, evaluated here by the hypercube-based diversity metric, it is possible to observe in the results shown in Tab. 1 that, as expected, cob-aiNet[MO] was the algorithm that presented the best diversity maintenance capabilities, obtaining significantly better results in problems *Deb & Tiwari*, *EBN* and *Lamé Supersphere*. For problem *Two-on-One 4*, KP1 (which was proposed with the same goals of cob-aiNet[MO] – diversity in the decision space) obtained the best average results.

Though the omni-optimizer presents diversity stimulation mechanisms devoted to the decision space, this algorithm was not able to obtain the best results w.r.t. diversity in any of the problems, although its results for *Two-on-One 4* were not significantly different from those of cob-aiNet[MO]. In order to visually illustrate the differences between cob-aiNet's and omni-optimizer's capability of finding and maintaining diversity in the decision space, Fig. 2 shows the final distribution of the solutions obtained by both algorithms in *Deb & Tiwari*'s problem. The results shown in Fig. 2 were obtained in the repetitions that led to the highest hypercube-based diversity for both algorithms. As it is possible to see, cob-aiNet[MO] was able to find and keep solutions in all the valleys of

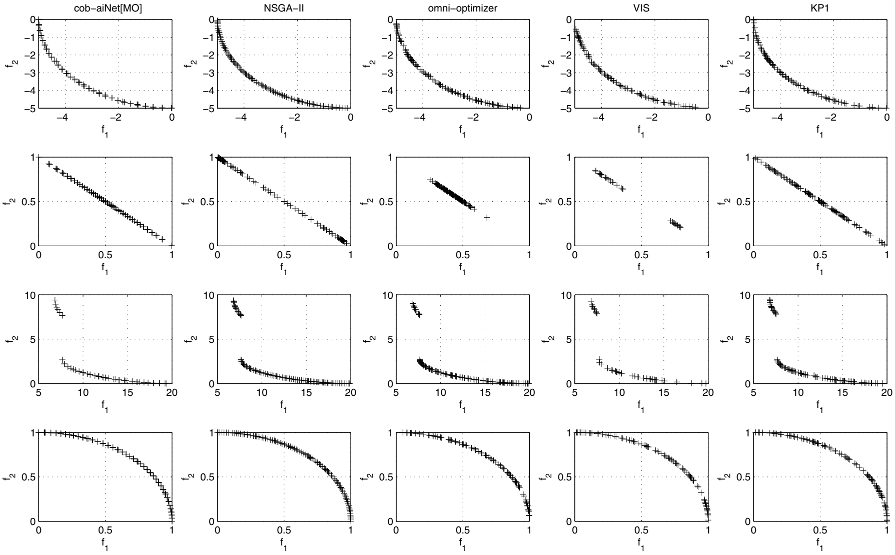


Fig. 1. Representation, in the objective space, of the final set of non-dominated solutions obtained in the repetition that led to the highest hypervolume for each algorithm. The rows correspond to problems *Deb & Tiwari* (first on the top), *EBN*, *Two-on-One 4* and *Lamé Supersphere*, respectively.

the problem, while the omni-optimizer (which obtained the second best performance for this problem) missed a few of these valleys. Due to space constraints, graphical results for the other three problems will be omitted here, but situations similar to that represented in Fig. 2 were also observed for them.

Table 1 also presents the overall average rank of each algorithm for each problem, according to all the metrics. This rank summarizes the results and indicates that, on average, cob-aiNet[MO] was slightly superior to the contenders.

One last result that is important to highlight is the overall computational cost of cob-aiNet[MO], shown in Tab. 2 together with the average computational time of NSGA-II (the fastest algorithm studied here). Immune-network-based algorithms generally tend to present higher computational costs when compared to other evolutionary approaches, as they adopt mechanisms that frequently require pairwise comparisons among all the solutions in their populations. In cob-aiNet[MO], this aspect is even more evident, as its concentration model and selection/insertion operator (see Sect. 2) require that these pairwise comparisons are made even more frequently. However, this extra computational cost, which can be seen as "the price to be paid" for the superior diversity maintenance capabilities and self-regulation of the population size, is independent of the *function evaluation cost*. So, the overall time required by both algorithms tend to be similar in many practical problems, in which function evaluations are generally the most computationally expensive operation.

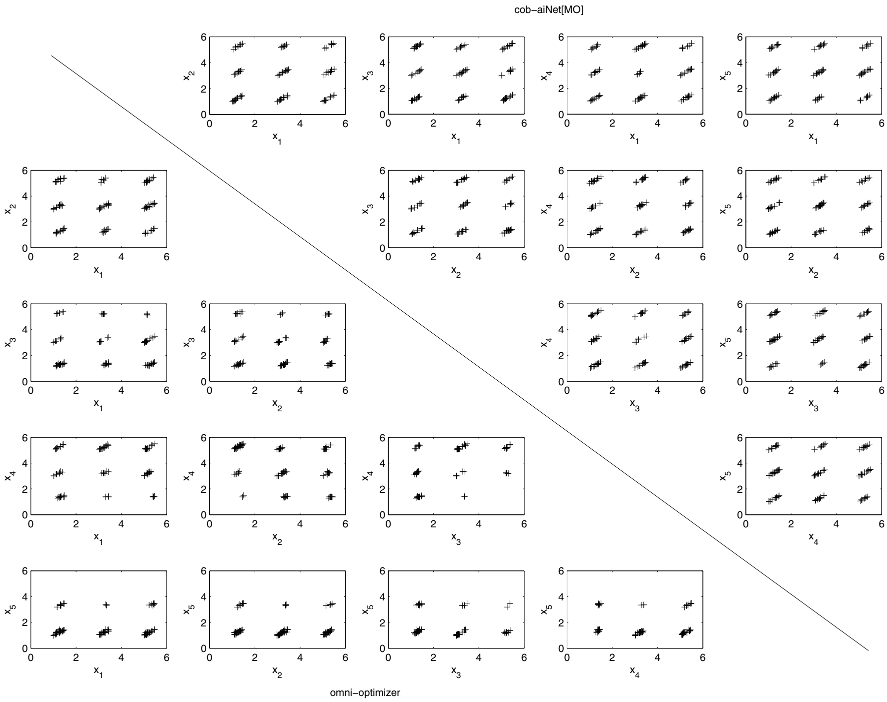


Fig. 2. Representation in the objective space of the final set of non-dominated solutions obtained by cob-aiNet[MO] (upper right triangle) and omni-optimizer (lower left triangle) for *Deb & Tiwari*'s problem. These plots correspond to the final populations in the repetitions that led to the highest hypercube-based diversity.

Table 2. Average ± std. deviation of the computational time (in seconds) required by cob-aiNet[MO] and NSGA-II. Both algorithms were written in Microsoft's Visual C# .NET. The experiments were performed on a machine with an Intel Core2Duo T7500 (2.2Ghz) processor, 3GB of RAM, Windows 7 and .NET Framework 4.

	Deb & Tiwari	**EBN**	**Two-on-One 4**	**Lamé Supersphere**
cob-aiNet[MO]	39.5223 ± 7.6719	89.6127 ± 7.9261	67.4790 ± 10.7241	54.5087 ± 5.9621
NSGA-II	3.7393 ± 0.1515	4.0877 ± 0.3094	4.0443 ± 0.1690	3.7940 ± 0.2034

4 Final Comments and Future Steps

This work presented and extension of the Concentration-based Artificial Immune Network Algorithm – cob-aiNet [5] – for multi-objective optimization problems. This extension, named cob-aiNet[MO], was proposed to tackle the not-so-well explored niche of multimodal multi-objective optimization problems, in which the maintenance of *diversity in the decision space* is often desirable.

The new algorithm was applied to four benchmark problems so that its capabilities to find and maintain a diverse set of non-dominated solutions could be evaluated, and it was compared to four algorithms from the literature: NSGA-II [7], omni-optimizer [8], VIS [12] and the proposal of Chan & Ray [4]. The obtained results have shown that cob-aiNet[MO] is a competitive algorithm, as it was able to obtain sets of solutions with the highest degree of diversity in three of the four problems studied here, and it was also capable of obtaining good results when the hypervolume and spread metrics were considered.

As future steps in this research, the cob-aiNet[MO] algorithm will be applied to a set of benchmark problems of higher dimension and higher number of objectives, so that its performance in more demanding scenarios can be properly evaluated. Other performance metrics will be applied to the obtained results, to allow a deeper comprehension of characteristics of the algorithm other than just its diversity maintenance capabilities. And finally, a sensitivity analysis will also be made, to evaluate and identify which parameters of the algorithm have the highest impact on its performance.

Acknowledgements

The authors would like to thank CAPES and CNPq for the financial support.

References

1. Bersini, H.: Revisiting idiotypic immune networks. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 164–174. Springer, Heidelberg (2003)
2. Burnet, F.M.: Clonal selection and after. In: Bell, G.I., Perelson, A.S., Pimgley Jr., G.H. (eds.) Theoretical Immunology, pp. 63–85. Marcel Dekker Inc., New York (1978)
3. de Castro, L.N., Timmis, J.: Artificial Immune Systems: a New Computational Intelligence Approach. Springer, Heidelberg (2002)
4. Chan, K.P., Ray, T.: An evolutionary algorithm to maintain diversity in the parametric and the objective space. In: Proc. of the 2005 Intl. Conference on Computational Intelligence, Robotics and Autonomous Systems, CIRAS 2005 (2005)
5. Coelho, G.P., Von Zuben, F.J.: A concentration-based artificial immune network for continuous optimization. In: Proc. of the 2010 IEEE International Conference on Evolutionary Computation, CEC 2010, pp. 108–115 (2010)
6. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
7. Deb, K., 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)
8. Deb, K., Tiwari, S.: Omni-optimizer: A procedure for single and multi-objective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 47–61. Springer, Heidelberg (2005)

9. 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)
10. Emmerich, M.T.M., Deutz, A.H.: Test problems based on Lamé superspheres. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 922–936. Springer, Heidelberg (2007)
11. de França, F.O., Coelho, G.P., Castro, P.A.D., Von Zuben, F.J.: Conceptual and practical aspects of the aiNet family of algorithms. Intl. Journal of Natural Computing Research 1(1), 1–35 (2010)
12. Freschi, F., Repetto, M.: VIS: An artificial immune network for multi-objective optimization. Engineering Optimization 38(8), 975–996 (2006)
13. Hart, E., Bersini, H., Santos, F.C.: How affinity influences tolerance in an idiotypic network. Journal of Theoretical Biology 249(3), 422–436 (2007)
14. Jerne, N.K.: Towards a network theory of the immune system. Annales d'Immunologie 125(1-2), 373–389 (1974)
15. Moore, D.S., McCabe, G.P., Craig, B.: Introduction to the Practice of Statistics, 6th edn. W.H. Freeman, New York (2007)
16. Preuss, M., Naujoks, B., Rudolph, G.: Pareto set and EMOA behavior for simple multimodal multiobjective functions. In: Runarsson, T., Beyer, H.G., Burke, E., Merelo-Guervós, J., Whitley, L., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 513–522. Springer, Heidelberg (2006)
17. Rudolph, G., Naujoks, B., Preuss, M.: Capabilities of EMOA to detect and preserve equivalent pareto subsets. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 36–50. Springer, Heidelberg (2007)
18. Shir, O.M., Preuss, M., Naujoks, B., Emmerich, M.: Enhancing decision space diversity in evolutionary multiobjective algorithms. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 95–109. Springer, Heidelberg (2009)
19. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221–248 (1994)
20. Toffolo, A., Benini, E.: Genetic diversity as an objective in multi-objective evolutionary algorithms. Evolutionary Computation 11(2), 151–167 (2003)
21. Unrich, T., Bader, J., Zitzler, E.: Integrating decision space diversity into hypervolume-based multiobjective search. In: Proc. of the 2010 Genetic and Evolutionary Computation Conference, GECCO 2010 (2010)
22. Yildiz, A.R.: A new design optimization framework based on immune algorithm and Taguchi's method. Computers in Industry 60(8), 613–620 (2009)
23. Zhou, A., Zhang, Q., Jin, Y.: Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 13(5), 1167–1189 (2009)
24. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm. Tech. rep., ETH, TIK, Zurich, Switzerland (2001)

Bi-objective Portfolio Optimization Using a Customized Hybrid NSGA-II Procedure

Kalyanmoy Deb[1], Ralph E. Steuer[2], Rajat Tewari[3], and Rahul Tewari[4]

[1] Department of Mechanical Engineering,
Indian Institute of Technology Kanpur, Kanpur, PIN 208016, India
deb@iitk.ac.in
Also Department of Business Technology, Aalto University,
School of Economics, Helsinki, Finland
[2] Department Banking and Finance, Terry College of Business,
University of Georgia, Athens, Georgia 30602-6253, USA
rsteuer@uga.edu
[3] Benaras Hindu University, Benaras, India
rajat.tewari.min08@itbhu.ac.in
[4] Deutsche Bank Group, Mumbai, India
rtewari.iitk@gmail.com

Abstract. Bi-objective portfolio optimization for minimizing risk and maximizing expected return has received considerable attention using evolutionary algorithms. Although the problem is a quadratic programming (QP) problem, the practicalities of investment often make the decision variables discontinuous and introduce other complexities. In such circumstances, usual QP solution methodologies can not always find acceptable solutions. In this paper, we modify a bi-objective evolutionary algorithm (NSGA-II) to develop a customized hybrid NSGA-II procedure for handling situations that are non-conventional for classical QP approaches. By considering large-scale problems, we demonstrate how evolutionary algorithms enable the proposed procedure to find fronts, or portions of fronts, that can be difficult for other methods to obtain.

1 Introduction

Portfolio optimization inherently involves conflicting criteria. Among possible objectives, minimizing risk and maximizing expected return (also known as the mean-variance model of Markowitz [1]) are the two objectives that have received the most attention. The decision variables in these problems are the proportions of initial capital to be allocated to the different available securities. Such bi-objective problems give rise to fronts of trade-off solutions which must be found to investigate the risk-return relationships in a problem.

In addition to the two objectives, these problems possess constraints [2]. The calculation of expected return is a linear function of the decision variables and the calculation of portfolio risk involves a quadratic function of the decision variables. Thus, the overall problem, in its simplest form, is a bi-objective quadratic

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 358–373, 2011.

programming (QP) problem, and when all constraints are linear and all variables are continuous, such problems can be solved exactly using the QP solvers [2,3]. However, in practice, there can be conditions which make QP solvers difficult to apply. For instance, a portfolio with very small investments in one or more securities is likely not to be of interest on managerial grounds alone. This then creates a need for decision variables that either take on a value of zero or a non-zero value corresponding to at least a minimum investment amount. Moreover, users may only be interested in portfolios that involve limited numbers of securities. Also, there can be governmental restrictions contributing to the complexity of the process, thus only adding to the difficulties of handling portfolio problems by classical means.

Evolutionary multi-objective optimization (EMO) has, over the years, been found to be useful in solving different optimization problems. EMO has also been used to solve different portfolio optimization problems [2,4,5,6,7]. Chang et al. [8] used genetic algorithms (GAs), tabu search, and simulated annealing on a portfolio optimization problem with a given cardinality on the number of assets. Other approaches including simulated annealing [9], differential evolution [10], and local search based memetic algorithms [11,12] have also been attempted. Here, we suggest and develop a customized hybrid NSGA-II procedure which is particularly designed to handle the non-smooth conditions alluded to above. The initialization procedure and the recombination and mutation operators are all customized so that the proposed procedure starts with a feasible population and always creates only feasible solutions. Conflicting objectives are handled using the elitist non-dominated sorting GA or NSGA-II [13]. To make the obtained solutions close to optimal solutions, the NSGA-II solutions are clustered into small groups and then a local search procedure is commenced from each solution until no further improvements are possible.

The remainder of the paper is organized as follows. Section 2 discusses portfolio optimization in greater detail. Thereafter, the customized hybrid NSGA-II procedure for solving portfolio optimization problems is described in Section 3. The local search component of the overall approach is described next. Section 4 presents results obtained by the proposed procedure. Concluding remarks constitute Section 5.

2 Practical Portfolio Optimization

In a portfolio problem with an asset universe of n securities, let x_i $(i = 1, 2, \ldots, n)$ designate the proportion of initial capital to be allocated to security i. There are typically two goals – maximize portfolio expected return and minimize portfolio risk [1]. To achieve the first goal, one might be tempted to pursue the securities with the highest individual expected returns r_i, but these are often the riskiest securities. In its most basic form, this results in the following problem:

$$\begin{aligned} \text{Minimize } & f_1(\mathbf{x}) = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} x_i \sigma_{ij} x_j, \\ \text{Maximize } & f_2(\mathbf{x}) = \textstyle\sum_{i=1}^{n} r_i x_i, \\ \text{subject to } & \textstyle\sum_{i=1}^{n} x_i = 1, \\ & x_i \geq 0. \end{aligned} \tag{1}$$

The first objective is portfolio risk as computed from a given $n \times n$ risk matrix $[\sigma_{ij}]$. The second objective is portfolio expected return as computed from a weighted sum of the individual security expected returns. The first constraint ensures the investment of all funds. The second constraint ensures the non-negativity of each investment.

It is clear that the objectives are conflicting. Thus, the solution to the above is the set of all of the problem's Pareto-optimal solutions as this is precisely the set of all of the problem's contenders for optimality. One of the ways to solve for the front of (1) is to convert the problem into a number of single-objective problems, the popular way being to convert the expected return objective to a $\geq$ constraint as in the e-constraint formulation:

$$\begin{aligned} \text{Minimize } & f_1(\mathbf{x}) = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} x_i \sigma_{ij} x_j, \\ \text{subject to } & \textstyle\sum_{i=1}^{n} r_i x_i \geq R, \\ & \textstyle\sum_{i=1}^{n} x_i = 1, \\ & x_i \geq 0. \end{aligned} \tag{2}$$

With the above a QP problem, the idea is to solve it repetitively for many different values of R.

If the number of securities n is more than about 50, it can be expected that almost any solution of (2) will contain many securities at the zero level. That is, it is expected that for many i, $x_i^* = 0$. It is also expected that for at least some securities, x_i^* will be a very small number. However, to have a practical portfolio, very small investments in any security may not be desired and are to be avoided. Thus, there is the practicality that for any portfolio to be of interest, there is to be a lower limit on any non-zero investment. That is, either $x_i^* = 0$ (meaning no investment in the i-th security) or $x_i^* \geq \alpha$ (meaning that there is a minimum non-zero investment amount for the i-th security). Additionally, there may also an upper bound β, limiting the maximum proportion of investment to any security. Unfortunately, the solution of Equation 2 for any given R does not in general carry with it the guarantee that all such requirements will be satisfied.

In addition to the above, there a second practicality which we now discuss and it is that any portfolio on the Pareto-optimal front of (1) may contain any number of non-zero investments. Over this, a user may wish to exert control. To generate practical portfolios, a user may be interested in specifying a fixed number of investments or a range in the number of investments. This constraint is also known as the cardinality constraint and has been addressed by other researchers as well [3,12]. Taking both practicalities into account, we have the following bi-objective optimization problem:

$$\begin{array}{ll} \text{Minimize} & f_1(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i \sigma_{ij} x_j, \\ \text{Maximize} & f_2(\mathbf{x}) = \sum_{i=1}^{n} r_i x_i, \\ \text{subject to} & \\ \text{(1st constraint)} & \sum_{i=1}^{n} x_i = 1, \\ \text{(2nd constraint)} & x_i = 0 \text{ or } \alpha \le x_i \le \beta, \\ \text{(3rd constraint)} & d^{\min} \le d(\mathbf{x}) \le d^{\max}, \end{array} \tag{3}$$

where $\alpha > 0$ and $d(\mathbf{x})$ is given as follows:

$$d(\mathbf{x}) = \sum_{i=1}^{n} \begin{cases} 1, \text{ if } x_i > 0, \\ 0, \text{ if } x_i = 0. \end{cases} \tag{4}$$

2.1 Difficulties with Classical Methods

Classical QP solvers face difficulties in the presence of discontinuities and other complexities. While the 1st constraint is standard, the 2nd constraint requires an 'or' operation. While $x_i = 0$ or $x_i = \alpha$ is allowed, values between the two are not. This introduces discontinuities in the search space. The 3rd constraint involves a parameter d which is defined by a discontinuous function of the decision variables, given in Equation 4. It is the presence of the 2nd and 3rd constraints that makes the application of standard methodologies difficult. In the following section, we discuss the GA based methodology of this paper for dealing with constraints like these.

3 Customized Hybrid NSGA-II Procedure

Generic procedures for handling constraints may not be efficient in handling the constraints of this problem. Here, we suggest a customized hybrid NSGA-II procedure for handling the specific constraints of the portfolio optimization problem. The 1st constraint ensures that all variables added together become one. Most evolutionary based portfolio optimization methodologies suggest the use of random keys or a dummy variable (x_i) and repair the variable vector as $x_i \leftarrow x_i / \sum_{i=1}^{n} x_i$ to ensure satisfaction of this constraint. In the presence of the 1st constraint alone, this strategy of repairing a solution is a good approach, but when other constraints are present, the approach may not be adequate.

The 2nd constraint introduces an 'or' operation between two constraints representing two disjointed feasible regions. One approach suggested in the literature is to use an additional Boolean variable $\rho_i \in [0, 1]$ for each decision variable x_i, as in [3]:

$$\alpha \rho_i \le x_i \le \beta \rho_i.$$

When $\rho_i = 0$, the variable $x_i = 0$ (meaning no investment in the i-th security). But when $\rho_i = 1$, the variable x_i takes a value in the range $[\alpha, \beta]$. This is an excellent fix-up for the 'or' constraint, but it introduces additional variables into the optimization problem.

The 3rd constraint involves a discontinuous function of decision variables. If Boolean variables are used to take care of the 2nd constraint, the cardinality constraint can be replaced by the following:

$$d^{\min} \leq \sum_{i=1}^{n} \rho_i \leq d^{\max}.$$

The presence of equality and inequality constraints would make it difficult for a generic penalty function approach to find a feasible solution and solve the problem.

Here, we suggest a different approach which does not add any new variables, but instead considers all constraints in its coding and evaluation procedures. First, a customized initialization procedure is used to ensure creation of feasible solutions. Thereafter, we suggest recombination and mutation operators which also ensure the creation of feasible solutions.

3.1 Customized Initialization Procedure

To create a feasible solution, we randomly select an integer value for d from the interval $[d^{\min}, d^{\max}]$. Then we create a portfolio which has exactly d non-zero securities. That is, we randomly select d of the n variables and assign zeros to the other $(n - d)$ variables. Since the non-zero x_i values must lie within $[\alpha, \beta]$, we randomly create a number in the range of each of the non-zero variables. However, the random numbers for the non-zero variables may not sum to one, thereby violating the first constraint. To make the solution feasible, we repair the solution as follows. We map the non-zero variables to another set of non-zero variables in the range $[\alpha, \beta]$ in such a way that they sum to one. The mapping function is shown in Figure 1. If $\sum_{i=1}^{n} x_i = 1$, we have a feasible solution, else we follow the procedure described below.

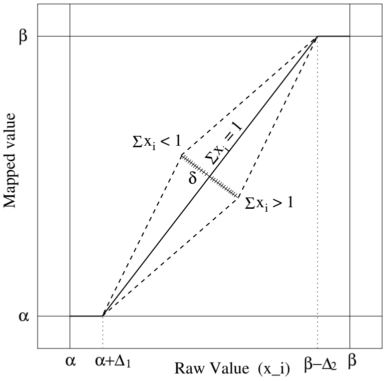


Fig. 1. The mapping function to satisfy the first constraint

1. If $\sum_{i=1}^{n} x_i \neq 1$, the decision vector $\mathbf{x}$ is modified as follows:
   - For each i, if x_i is within $[\alpha, \alpha + \Delta_1]$, set $x_i = \alpha$.
   - For each i, if x_i is within $[\beta - \Delta_2, \beta]$, set $x_i = \beta$.
   - For each i, if x_i is within $[\alpha + \Delta_1, \beta - \Delta_2]$, x_i is kept the same.
2. Compute $X = \sum_{i=1}^{n} x_i$. If X is still not equal to one, then stretch the mapping line by moving its mid-point by an amount δ, as follows:
   - If $X < 1$, move the mid-point upwards by a distance δ in a direction perpendicular to the line (as shown in Figure 1). This process is continued until $\sum_{i=1}^{n} x_i$ is arbitrarily close to one.

- If $X > 1$, move the mid-point downwards by a distance δ in a direction perpendicular to the line. This process is continued until $\sum_{i=1}^{n} x_i$ is arbitrarily close to one.

The parameters Δ_1 and Δ_2 allow us to map near-boundary values to their boundary values. However, if this is unwarranted, both of these parameters can be chosen to be zero, particularly in the cases when all x_i values are either within $[\alpha, \alpha + \Delta_1]$ or within $[\beta - \Delta_2, \beta]$. The parameter δ may be chosen to be a small value and it helps to iterate the variable values so that the 1st constraint is satisfied within a small margin. A small value of δ will require a large number of iterations to satisfy the first constraint but the difference between $\sum_{i=1}^{n} x_i$ and one will be small and vice versa. It remains as an interesting study to establish a relationship between δ and the corresponding error in satisfying the equality constraint, but here, in all of our simulations, we have arbitrarily chosen the following values: $\Delta_1 = 0$, $\Delta_2 = 0.001$ and $\delta = 10^{-6}$.

After this repair mechanism, the 1st constraint is expected to be satisfied within a small tolerance. The 2nd and 3rd constraints are also satisfied in the process. Thus, every solution created by the above process is expected to be feasible.

3.2 Customized Recombination Operator

Once a feasible initial population is created, it is to be evaluated and then recombination and mutation operators are to be applied. Here, we suggest a recombination operator which always produces feasible offspring solutions by recombining two feasible parent solutions.

Two solutions are picked at random from the population as parents. Let the number of non-zero securities in the parents be d_1 and d_2, respectively. We illustrate the recombination procedure through the following two parent solutions having $n = 10$:

$$\begin{array}{l} \text{Parent1: } a_1\ 0\ a_3\ \ 0\ \ a_5\ 0\ \ 0\ \ a_8\ 0\ a_{10} \\ \text{Parent2: } b_1\ 0\ b_3\ b_4\ b_5\ 0\ b_7\ \ 0\ \ 0\ b_{10} \end{array}$$

It is clear that $d_1 = 5$ and $d_2 = 6$ in the above. The following steps are taken to create a child solution.

1. An integer d_c is randomly selected from the range $[d_1, d_2]$, say, for the above parents, $d_c = 5$.
2. The child solution inherits a zero investment for a particular security if both parents have zero investments in that security. For the above parents, this happens for three (n_0) securities: $i = 2$, 6 and 9. Thus, the partial child solution is as follows:

$$\text{Child1: } - \ 0 - - - \ 0 - - \ 0 -$$

3. The securities which have non-zero investments in both parents will inherit a non-zero investment value, but the exact amount will be determined by a

real-parameter recombination operator (bounded form of SBX [14]) applied to the parent values. This operator ensures that values are not created outside the range $[\alpha, \beta]$. We discuss this procedure a little later. For the above parents, the child solution then takes on the form:

$$\text{Child1: } c_1 \; 0 \; c_3 - c_5 \; 0 - - \; 0 \; c_{10}$$

Note that the number of common non-zero investments (n_c) between the two parent solutions is such that $n_c \in [0, \min(d_1, d_2)]$. In this case, $n_c = 4$.

4. The number of slots that remain to be filled with non-zero values is $w = d_c - n_c$. Since, $d_c \in [d_1, d_2]$, w is always greater than or equal to zero. From the remaining $(n - n_0 - n_c)$ securities, we choose w securities at random and take the non-zero value from the parent to which it belongs. For the above example parents, $w = 5-4 = 1$ and there are $(10-3-4)$ or 3 remaining places to choose the $w = 1$ security from. The remaining securities are assigned a value zero. Say, we choose the seventh security to have a non-zero value. Since the non-zero value occurs in parent 2, the child inherits b_7. Thus, the child solution looks like the following:

$$\text{Child1: } c_1 \; 0 \; c_3 \; 0 \; c_5 \; 0 \; b_7 \; 0 \; 0 \; c_{10}$$

5. The above child solution is guaranteed to satisfy both 2nd and 3rd constraints, but needs to satisfy the 1st constraint. We use the procedure described in Subsection 3.1 to repair the solution to a feasible one.

We also create a second child from the same pair of parents using the same procedure. Due to the creation of the random integer d_c and other operations involving random assignments, the second child is expected to be different from the first child. However, both child solutions are guaranteed to be feasible.

The bounded form of the SBX operator is described here for two non-zero investment proportions for a particular security (say, a_1 and b_1, taken from parents 1 and 2, respectively). The operator requires a user-defined parameter η_c:

1. Choose a random number u_i within $[0, 1]$.
2. Calculate γ_{qi} using the equation:

$$\gamma_{qi} = \begin{cases} (\kappa u_i)^{\frac{1}{\eta_c+1}}, & \text{if } u_i \leq \frac{1}{\kappa}, \\ \left(\frac{1}{2-\kappa u_i}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases} \tag{5}$$

For $a_1 \leq b_1$, $\kappa = 2 - \zeta^{-(\eta_c+1)}$ and ζ is calculated as follows: $\zeta = 1 + 2\min[(a_1 - \alpha),\ (\beta - b_1)]/(b_1 - a_1)$. For $a_1 > b_1$, the role of a_1 and b_1 can be interchanged and the above equation can be used.

3. Then, compute the child from the following equation:

$$c_1 = 0.5[(1 + \gamma_{qi})a_1 + (1 - \gamma_{qi})b_1]. \tag{6}$$

The above procedure allows a zero probability of creating any child solutions outside the prescribed range $[\alpha, \beta]$ for any two solutions a_1 and b_1 within the same range.

3.3 Customized Mutation Operator

Mutation operators perturb the variable values of a single population member. We use two mutation operators.

Mutation 1. In this operation, a non-zero security value is perturbed in its neighborhood by using the polynomial mutation operator [15]. For a particular value a_1, the following procedure is used with a user-defined parameter η_m:

1. Create a random number u within $[0, 1]$.
2. Calculate the parameter $\bar{\mu}$ as follows:

$$\bar{\mu} = \begin{cases} (2u)^{\frac{1}{\eta_m+1}} - 1, & \text{if } u \leq 0.5, \\ 1 - [2(1-u)]^{\frac{1}{\eta_m+1}}, & \text{otherwise.} \end{cases} \quad (7)$$

3. Calculate the mutated value, as follows:

$$a_1' = a_1 + \bar{\mu} \min[(a_1 - \alpha),\ (\beta - a_1)].$$

The above procedure ensures that a_1' lies within $[\alpha, \beta]$ and values close to a_1 are preferred more than values away from a_1. The above procedure is applied to each non-zero security with a probability p_m.

Since values are changed by this mutation operator, the repair mechanism of Subsection 3.1 is applied to the mutated solution.

Mutation 2. The above mutation procedure does not convert a zero value to a non-zero value. But, by the second mutation operator, we change z zero securities to non-zero securities, and to keep the number of investments d the same, we also change z non-zero securities to zero securities. For this purpose, a zero and a non-zero security are chosen at random from a solution and their values are swapped. The same procedure is re-applied to another $(z-1)$ pairs of zero and non-zero securities. To illustrate, we set $z = 2$ and apply the operator on the following solution:

$$a_1\ 0\ a_3\ 0\ a_5\ 0\ 0\ a_8\ 0\ a_{10}$$

Mutation 2 operator is applied between the first and fourth securities and between the ninth and tenth securities as follows:

$$0\ 0\ a_3\ a_1\ a_5\ 0\ 0\ a_8\ a_{10}\ 0$$

Since no new non-zero values are created in this process, the 1st constraint is always be satisfied and this mutation operator always preserves the feasibility of a solution. In all our simulations, we have chosen $z = 2$, although other values of z may also be tried.

Thus, the proposed procedure starts with a set of feasible solutions. Thereafter, the creation of new solutions by recombination and mutation operators always ensures their feasibility. These procedures help NSGA-II to converge close to the Pareto-optimal front trading off the two conflicting objectives quickly.

3.4 Clustering

NSGA-II is expected to find a set of solutions representing the trade-off frontier. The NSGA-II procedure requires a population size (N) proportional to the size of the problem (number of securities) for a proper working of its operators. However, this can cause NSGA-II to find more portfolios than a user may desire. To reduce the number of obtained trade-off portfolios, we apply a k-mean clustering procedure to the obtained set of solutions and pick only H ($< N$) well-distributed portfolios from the obtained NSGA-II solutions. For all our problems, we use $H = 30$ to get an idea of the trade-off frontier, however in practice other numbers of solutions can be employed.

3.5 Local Search

To obtain near-Pareto-optimal solutions, we also employ a local search procedure from all H clustered solutions. We use a single-objective GA with the above-discussed operators for this purpose. From a seed (clustered) solution $\mathbf{z}$, first an initial population is created by mutating the seed solution N_{ls} times (where N_{ls} is the population size for the local search). Both mutation operators discussed above are used for this purpose. The mutation probability p_m and parameter η_m are kept the same as before. Since the mutated solutions are repaired, each population member is guaranteed to be feasible. The objective function used here is the following achievement scalarizing function [16]:

$$F(\mathbf{x}) = \min_{j=1}^{2} \frac{f_j(\mathbf{x}) - z_j}{f_j^{\max} - f_j^{\min}} + 10^{-4} \sum_{j=1}^{2} \frac{f_j(\mathbf{x}) - z_j}{f_j^{\max} - f_j^{\min}}. \qquad (8)$$

Here, $f_2(\mathbf{x})$ is considered to be the negative of the expected return function (second objective) given in Equation 3. The parameters $f_j^{\min}$ and $f_j^{\max}$ are the minimum and maximum objective values of all obtained NSGA-II solutions. In the local search GA, the above function is minimized. The search continues until no significant improvement in consecutive population-best solutions is obtained.

4 Results

We are now ready to present some results using our proposed customized hybrid NSGA-II procedure. For simulation, we consider a data-set (containing an $\mathbf{r}$ expected return vector and a $[\sigma_{ij}]$ risk matrix) for a 1,000-security problem. The data-set was obtained from the random portfolio-selection problem generator specified in [17]. Figure 2 shows the risk-return trade-off frontier first obtained by solving the e-constraint problem (Equation 2) many times via QP for different values of R. Solutions that have low risk (near A) also have low expected returns. Solutions that have high risk (near C) also have a high expected returns. Although this is a known fact, the discovery of the trade-off frontier provides the actual manner in which the trade-offs take place. For example, for the risk

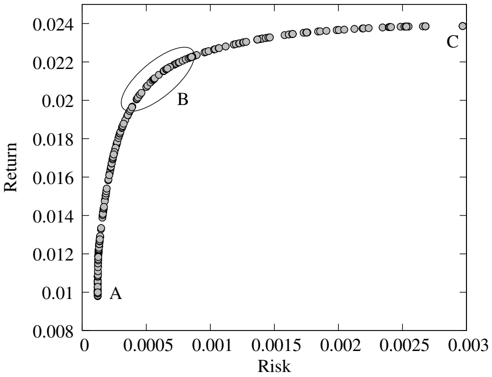


Fig. 2. Obtained trade-off front of the original problem using the quadratic programming (QP) technique

matrix and return vector chosen for this case study there seem to be good trade-offs near B, and a most preferred portfolio may well be selected from the region marked. This is because to choose a solution near A over a solution from B, one has to sacrifice a large amount of expected return to get only small reduction in risk. Similarly, to choose a solution C over a solution from B, one has to take on a large amount of extra risk to get a only small increase in expected return. However, since not all trade-off frontiers are so nicely formed and who is to know from where along a trade-off frontier a given decision maker will make a final selection, it is not our intention to become involved in decision making at this point. Rather, our goal here is to develop comprehensive sets of trade-off frontier portfolios in problems involving practicalities so that no region possibly containing an optimal portfolio will be missed in the analysis.

For this case study, we analyzed all solutions and observed that out of the 1,000 securities, only 88 ever took on non-zero investments. We extract for these securities their **r**-vector and $[\sigma_{ij}]$ risk matrix information and confine our further studies to only these $n = 88$ securities.

For the NSGA-II procedure, we selected a population size of $N = 500$, a maximum number of 3,000 generations, a recombination probability of $p_c = 0.9$, a SBX index of $\eta_c = 10$, a mutation probability of $p_m = 0.01$, and a polynomial mutation index of $\eta_m = 50$.

For the local search procedure, we selected $N = 200$ and the ran until 500 generations were reached. All other GA parameters were kept the same as in the NSGA-II simulations. It is important to mention here that since NSGA-II considers all constraints given in Equation 3 and QP considered only the first constraint, the NSGA-II solutions are not expected to be better than QP solutions. But NSGA-II solutions are expected to be non-dominated to those QP solutions that satisfy all constraints given in Equation 3.

4.1 Portfolio Optimization with No Size Constraint

As the 1st constraint is always present, we start by imposing the 2nd constraint with $\alpha = 0.005$ and $\beta = 0.04$. Of the 483 QP obtained portfolios characterizing the front of the problem without any discontinuities, 453 satisfy the bounds of the 2nd constraint. Due to the use of $\beta = 0.04$, at least 25 securities, of course, are in every portfolio. However, in our algorithm presented, we do not specify any restriction on the number of securities.

Figure 3 shows 294 trade-off solutions obtained by the customized NSGA-II procedure alone. The QP solutions that satisfy the $\alpha = 0.005$ and $\beta = 0.04$ bounds are also shown in the figure. The figure clearly shows that the proposed procedure is able to find a good set of distributed solutions close to the points obtained by the QP procedure. High risk solutions are somewhat worse than those obtained by QP. But we show shortly that the hybrid method in which a local search is applied on the NSGA-II solutions is a better overall strategy.

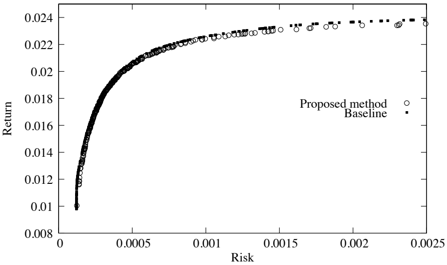


Fig. 3. Obtained trade-off front for $\alpha = 0.005$ and $\beta = 0.04$ and with no restriction on d by a typical NSGA-II run

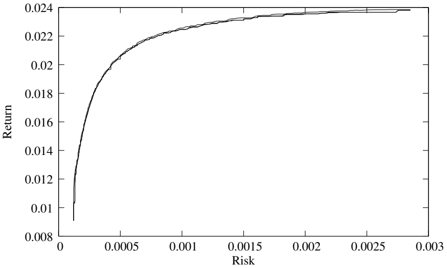


Fig. 4. Attainment curve of the 21 runs for this problem for the case $\alpha = 0.005$ and $\beta = 0.04$ and with no restriction on d

Genetic algorithms are stochastic optimization procedures and their performance is shown to depend somewhat on the chosen initial population. In order to demonstrate the robustness of the proposed customized NSGA-II procedure over variations in the initial population, we created 21 different random initial populations and ran the proposed procedure independently from each one. The obtained solutions are collected together and 0%, 50% and 100% attainment curves [18] are plotted in Figure 4. The 0% attainment curve indicates the trade-off boundary which dominates all obtained trade-off solutions. The 100% attainment curve indicates the trade-off boundary that is dominated by all obtained trade-off solutions collectively. The 50% attainment curve is that trade-off boundary that dominates 50% of the obtained trade-off solutions. The three curves obtained for the 21 non-dominated fronts are so close to each other that they cannot be well distinguished visually. This means that all 21 independent runs produce almost identical frontiers thereby indicating the reliability of the proposed customized NSGA-II procedure.

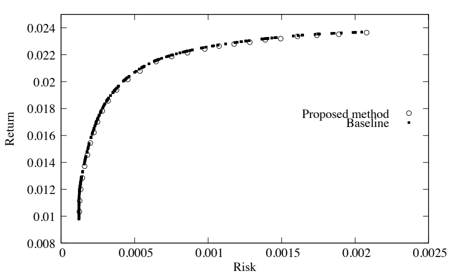


Fig. 5. A set of 30 trade-off portfolios obtained using clustering and local search procedure for the case with $\alpha = 0.005$, $\beta = 0.04$, and no restriction on d

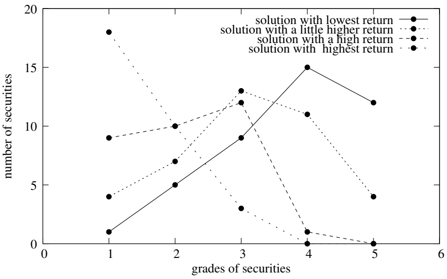


Fig. 6. Investment pattern for four solutions in each of the five grades of diminishing return

From the obtained non-dominated front of all 21 runs, we now apply the clustering algorithm and choose only 30 well-distinguished portfolios covering the entire front. These solutions are then local searched. The complete method is called the customized hybrid NSGA-II procedure. The obtained solutions are shown in Figure 5 by circles. The solutions are close to the QP solutions, but importantly a well-distributed set of portfolios is created by our proposed procedure.

In order to show the investment pattern of different trade-off portfolios, we choose two extreme solutions and two intermediate solutions from the obtained set of 30 portfolios. Based on the return values r_i, all securities are ordered from the highest return value to the lowest. They are then divided into five grades of 18 securities each, except for the fifth grade which has the lowest 16 securities. For each of the four solutions, the number of non-zero securities in each grade is counted and plotted in Figure 6. It can be seen that the highest return solution allocates the maximum number of grade 1 securities in order to maximize the overall return. The solution with the lowest return (or lowest risk) invests more into grade 5 securities (having lowest return). The intermediate solutions invest more into intermediate grades to make a good compromise between return and risk.

4.2 Portfolio Optimization for a Fixed d

Now we impose the 3rd constraint. Initial population members are guaranteed to satisfy the 3rd constraint and every recombination and mutation operation also guarantees maintaining the 3rd constraint.

First, we consider $d^{\min} = d^{\max} = 28$, so that portfolios with only 28 securities are desired. The variable bounds of $\alpha = 0.005$ and $\beta = 0.04$ are enforced. The QP generated frontier now has only 24 portfolios which satisfy these constraints. They are shown in Figure 7. There are a total of 168 portfolios found by a typical run of the proposed procedure. They are shown in the figure as well. It is clear from the figure that the proposed procedure is able to find widely distributed

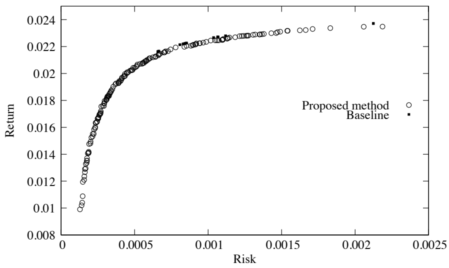


Fig. 7. Obtained trade-off front for $d = 28$ and with $\alpha = 0.005$ and $\beta = 0.04$ by a typical NSGA-II run

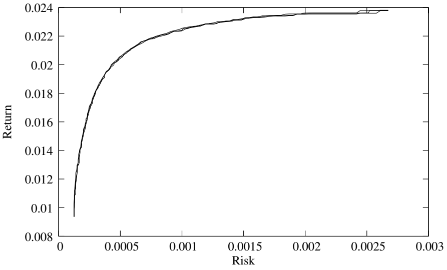


Fig. 8. Attainment surface of the 21 runs for the problem with $d = 28$ and $\alpha = 0.005$ and $\beta = 0.04$

sets of solutions. In order to investigate the robustness of the customized NSGA-II procedure alone, the three attainment curves are drawn for the 21 different runs in Figure 8, as before. The closeness of these curves to one another signifies that the proposed procedure is able to find almost identical trade-off frontiers on any run.

Finally, all solutions from 21 runs are collected together and non-dominated solutions are clustered into 30 widely separated solutions. They are local searched and shown in Figure 9. In this case, it is clear from the figure that the proposed procedure is able to find much wider and better distributed sets of portfolios than classical QP alone.

We consider another case in which exactly $d = 50$ securities are desired in each portfolio. The QP solution set did not have any such portfolio. When we apply the customized hybrid NSGA-II procedure, we obtain the front with 30 portfolios shown in Figure 10. These results amply demonstrate the working of the proposed procedure when both a discontinuous variable bound and a discrete number of securities are desired.

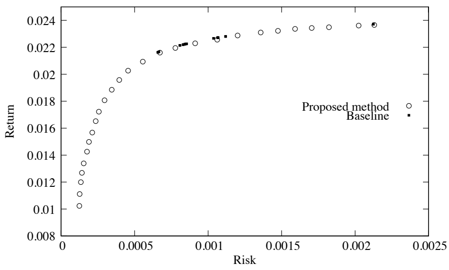


Fig. 9. A set of 30 trade-off portfolios obtained using clustering and local search procedure for the case with $d = 28$, $\alpha = 0.005$, and $\beta = 0.04$

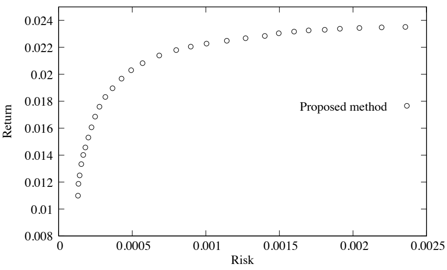


Fig. 10. 30 different portfolios found for $d = 50$, $\alpha = 0.005$, and $\beta = 0.04$ obtained using the proposed procedure. Base-line QP solution set did not have any solution with $d = 50$.

4.3 Portfolio Optimization for a Given Range of *d*

Next, we consider the case in which a user wishes the portfolios to vary within a prescribed range $[d^{\min}, d^{\max}]$. Customized operators to handle such variables have been described before. We consider $d^{\min} = 30$ and $d^{\max} = 45$ and the variable bounds $\alpha = 0.005$ and $\beta = 0.04$. The distribution of solutions obtained after local search is shown in Figure 11. There are 261 QP obtained solutions that satisfy these constraints. The closeness of the obtained solutions with the QP solutions is clear from the figure.

In order to understand the investment pattern, we consider 50 clustered solutions from the set of all 21 runs and plot the number of non-zero securities in each solution with the corresponding risk value (objective $f_1(\mathbf{x})$) in Figure 12. It is interesting to note that low risk portfolios (also with low returns) can be obtained with a wide variety of investments, almost uniformly from 30 to 44 securities. However, high risk (therefore, high return) portfolios only require users to invest in smaller numbers of securities.

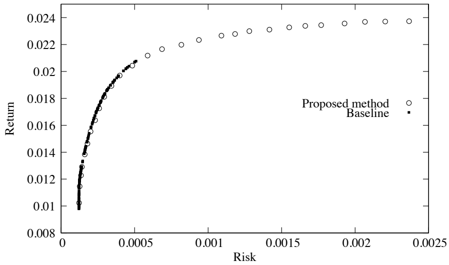


Fig. 11. A set of 30 portfolios obtained for $d \in [30, 45]$, $\alpha = 0.005$, and $\beta = 0.04$

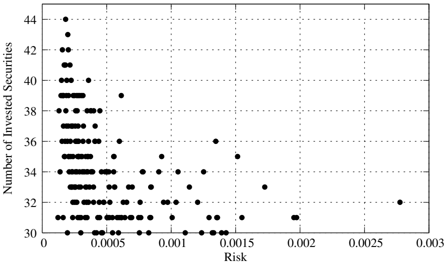


Fig. 12. Number of invested securities (d) versus accumulated risk

5 Conclusions

In this paper, we have suggested a customized hybrid NSGA-II procedure for handling a bi-objective portfolio optimization problem having practicalities. First, the proportion of investment in a security can either be zero (meaning no investment) or be between specified minimum and maximum bounds. Second, the user may have a preference for a certain number, or a range in the number, of non-zero securities. Such requirements make the resulting optimization problem difficult to solve via classical techniques. The hallmark of our study is that our procedure uses a repair mechanism to make all generated solutions feasible. By taking a large-sized problem, we have systematically demonstrated the efficacy of the proposed procedure over different user-defined requirements. The reliability of the proposed customized hybrid NSGA-II procedure is demonstrated by simulating the algorithm from 21 different initial populations. The accuracy of the proposed procedure is enhanced by coupling the method with a clustering and a local search procedure. The results are compared with those obtained with the quadratic programming method.

The proposed methodology involves a number of parameters (Δ_1, Δ_2, δ, z). A parametric study is now needed to find suitable values of these parameters. A computational complexity study of the proposed repair mechanism is also needed to establish a relationship between δ and extent of violation of the equality (1st) constraint. The methodology suggested here is now ready to be applied to more complex portfolio optimization problems. The QP method, when applicable, is generally the best approach to pursue. But practice is full of complexities which can limit the use of standard QP methods. The flexibility of genetic algorithms demonstrated here provides us with confidence about their applicability to real-life portfolio optimization problems. This study should encourage further use of QP and GA approaches together in a probably more computationally efficient manner to make the overall portfolio optimization problem more pragmatic.

References

1. Markowitz, H.M.: Portfolio selection: Efficient diversification of investments. Yale University Press, New York (1959)
2. Steuer, R.E., Qi, Y., Hirschberger, M.: Suitable-portfolio investors, nondominated frontier sensitivity, and the effect of multiple objectives on standard portfolio selection. Annals of Operations Research 152(1), 297–317 (2007)
3. Stein, M., Branke, J., Schmeck, H.: Efficient implementation of an active set algorithm for large-scale portfolio selection. Computers & Operations Research 35(12), 3945–3961 (2008)
4. Zhang, W.-G., Chen, W., Wang, Y.-L.: The adaptive genetic algorithms for portfolio selection problem. IJCSNS International Journal of Computer Science and Network Security 6(1), 196–200 (2006)
5. Lin, D., Li, X., Li, M.: A genetic algorithm for solving portfolio optimization problems with transaction costs and minimum transaction lots. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 808–811. Springer, Heidelberg (2005)
6. Lin, C.-M., Gen, M.: An effective decision-based genetic algorithm approach to multiobjective portfolio optimization problem. Applied Mathematical Sciences 1(5), 201–210 (2007)
7. Branke, J., Scheckenbach, B., Stein, M., Deb, K., Schmeck, H.: Portfolio optimization and an envelope-based multi-objective evolutionary algorithm. European Journal of Operational Research 199(3), 684–693 (2009)
8. Chang, T.-J., Meade, N., Beasley, J.B., Sharaiha, Y.: Heuristics for cardinality constrained portfolio optimisation. Comput. & Op. Res. 27, 1271–1302 (2000)
9. Crama, Y., Schyns, M.: Simulated annealing for complex portfolio selection problems. European Journal of Operational Research 150, 546–571 (2003)
10. Krink, T., Paterlini, S.: Differential evolution for multiobjective portfolio optimization. Center for Economic Research (RECent) 021, University of Modena and Reggio E., Dept. of Economics (2008)
11. Streichert, F., Tanaka-Yamawaki, M.: A memetic algorithm on the constrained portfolio selection problem. Institute of Statistical Mathematics 187, 140–149 (2006)

12. Streichert, F., Tanaka-Yamawaki, M.: The effect of local search on the constrained portfolio selection problem. In: IEEE International Congress on Evolutionary Computing (CEC 2006), pp. 2368–2374. IEEE Press, Piscatway (2006)
13. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
14. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems 9(2), 115–148 (1995)
15. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)
16. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Applications, pp. 468–486. Springer, Heidelberg (1980)
17. Hirschberger, M., Qi, Y., Steuer, R.E.: Randomly generating portfolio-selection covariance matrices with specified distributional characteristics. European Journal of Operational Research 177(3), 1610–1625 (2007)
18. Fonseca, C.M., da Fonseca, V.G., Paquete, L.: Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 250–264. Springer, Heidelberg (2005)

Aesthetic Design Using Multi-Objective Evolutionary Algorithms

António Gaspar-Cunha[1], Dirk Loyens[2,3], and Ferrie van Hattum[1]

[1] Institute of Polymer and Composites/I3N, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
agc@dep.uminho.pt, fvh@dep.uminho.pt
http://www.dep.uminho.pt/agc/

[2] School of Architecture, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal

[3] ESAD Escola Superior de Artes e Design, Avenida Calouste Gulbenkian, 4460-268 Senhora da Hora, Matosinhos, Portugal
dirkloyens@gmail.com

Abstract. The use of computational methodologies for the optimization of aesthetic parameters is not frequent mainly due to the fact that these parameters are not quantifiable and are subjective. In this work an interactive methodology based on the use of multi-objective optimization algorithms is proposed. This strategy associates the results of different optimization runs considering the existent quantifiable objectives and different sets of boundary conditions concerning the decision variables, as defined by an expert decision maker. The associated results will serve as initial population of solutions for a final optimization run. The idea is that a more global picture of potential "good" solutions can be found. At the end this will facilitate the work of the expert decision maker since more solutions are available. The method was applied to a case study and the preliminary results obtained showed the potentially of the strategy adopted.

Keywords: aesthetic design, multi-objective evolutionary algorithms.

1 Introduction

Digital design culture and new paradigms in digital design thinking have a great impact on the design, development and realization of components and objects. Projects frequently embody a trade-off between multiple and interdependent requirements such as performance-related aspects, form freedom and complexity of the desired architectural expressions.

Current design methods, though already largely involving digital tools and processes, are not yet fully suited to dynamically optimize the design within its multiple boundary conditions. At the same time, conventional materials and technologies compromise the realization of the optimized design and its underlying concepts. Here, polymer and composite materials, in combination with their

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 374–388, 2011.

largely automated manufacturing methods, are a powerful group of materials to overcome this dilemma due to their inherent properties, such as aesthetically pleasing, lightness, ability to mould complex shapes, ease of fabrication and integration of parts.

In this research a computational method has been developed that can iteratively optimize a design towards its functional requirements using available design, simulation and user-interfacing tools. The method has been applied to the optimization of a 'generic' roof structure towards daylight conditions while minimizing area and thus weight and materials used. Multi-Objective Evolutionary Algorithms (MOEA) in combination with a decision making methodology have been used, together with critical Decision Maker (DM) interaction. The result indicates the usefulness of this model and the developed techniques in the early stage of the design process, leading to better design solutions.

This text is organized as follows: in section 2 the state-of-the-art concerning digital design methodology and the practical problem to be solved are presented; the problem characteristics as well the optimization methodology adopted are described in detail in section 3; in section 4 the methodology proposed is applied to an example and in section 5 the conclusions are stated.

2 Digital Design Method

2.1 State of the Art

Computers have been used in the design process for over fifty years. Initially the use of computers was limited to drawing, representation, basic structural analysis or construction planning. Eventually performance analysis was executed as an afterthought, but always as part of an essentially paper-based design process [1]. It was not until the moment when design moved away from the conventional logic of representation and instead started interacting with the process of form generation itself, which we can speak of a new paradigm in the field of digital design [2]. Since then digital design has evolved into a new and unique form of design.

The increasing integration of sophisticated and interactive digital design media throughout the complete design process, from early concept development until iterative testing and fabrication, has already provoked the emerging of new ways of design making and new ways of design thinking [3,4]. These concepts are starting to be the subject of research in the field of architecture. The concept of adaptation has been used to guide research towards the application of evolution-based generative design systems to shape architectural forms [5]. Other research has evolved in the development of specific software for methods combining structural grammars, performance metrics, structural analysis and stochastic optimization [6].

A compound model of digital design has been proposed as a future class of paradigmatic digital design [2]. These compound models are based on the integration of form finding, form generation, evaluation and performance processes.

The development of processes or methods that aim to create new collaborative relationships between designer and computer, based on the idea of continuous feedback, are appointed as desired future research topics, though this research area is largely still in its infancy [6,7].

Although eminent architectural objects are today present, that have only been possible through the use of digital tools, the design processes used do not completely and interactively optimise at an early design stage [8]. At the same time, realisation has been frequently limited to the use of more traditional construction materials.

A desired design method would go beyond this [8]. The method should adopt processes that allow for dynamical multi-objective design optimization, integrated with - but not inherently limited to a sub-set of - available, both 'off-the-shelf' and novel, material solutions. In construction process composite materials can play an important role, as they are known for their ability to combine the moulding of complex forms with varying, tailorable ranges of outstanding properties either aesthetic or structural, with a relative ease-of-processing and a large level of functional integration [9]. Due to these characteristics, they have been a favorite material in design prototyping and final object manufacture, offering possibilities unable to be embodied in other materials. Furthermore, their fabrication can be highly automated, thus allowing for a high level integration with the desired digital design environment [10].

2.2 Experimental

The current research project explores new digital paradigms in a project development process within a framework of design as information processing rather than simple form finding. The project explores new relationships between the designer-as-toolmaker, information, process and the object. In this way the potential distinctive character of digital design thinking will be explored.

A new method is developed and tested, allowing integrating of complex quantitative and qualitative requirements at an early stage in the design process. This is achieved by combining multiple digital performance simulation tools with algorithms with generative capabilities, acting in the fuzzy front end of conceptual development. In this way, the design process is quicker and with more iterations, allowing complex functional and performance requirement integration and posing almost no limit to the freedom and complexity of forms and components used. As a first step, the method is applied to the fields of design, engineering and architecture, demonstrating that the existing design computing technologies, available and readily used in fields of architecture and composite technology, can open new territories for conceptual exploration.

For this purpose a generic roof structure geometry, represented by a single surface, was taken as a starting point (see Figure 1). This roof structure is represented by a single Non-Uniform Rational Basis Spline (NURBS) surface [11]. This method allows for a precise mathematical representation of a free form surface and for precise control by manipulating the control points of the surface. The control points determine the shape of the curve and a single control point only influences those intervals where it is active. This allows for the changing of one part of a surface while keeping other parts equal. The manipulation of control points is used in the everyday meaning of the word 'point', a location in 3D space defined by its coordinates.

In the present study a set of 20 control points were defined, allowing the virtually unlimited adaptation of the surface geometry. Based on the set of spatial coordinates of the control points, the surface is built in a general 3D design software [12]. The area is calculated and the surface exported to a building analysis software [13] for subsequent numerical analysis, in this case the average daylight factor under the structure, as an indication of the light functionality of the structure. The results (Area, Daylight) are saved for subsequent use by the optimization routine, as described in the subsequent section.

The resulting optimized design combines both quantitative and qualitative evaluation of the design's performance, leading the exploration of a wider range of design solutions at an early stage in the concept phase. The best performing concept can then be used as the starting point for subsequent detailed design. The proposed model thus results in the streamlining of the design and development processes of architectural objects with a high degree of form freedom and system complexity. Applying this approach, architects and designers can conceive interactively, test the consequences of actions almost immediately, and explore different ways of solution refinements that are crucial in design and architecture.

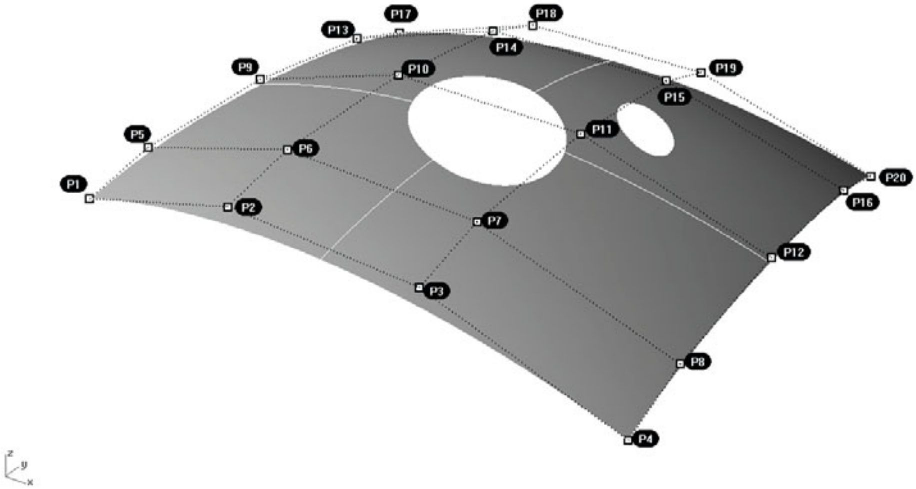


Fig. 1. Studied roof structure were the geometry is defined by the NURBS surface methodology (the limits for the coordinates of the control points are 5 meters)

3 Multi-objective Optimization

3.1 Problem Characteristics

As mentioned above the problem to be solved has three objectives to be accomplished, the minimization of both Area and Daylight and the aesthetics design. The minimization of the structure Area, a measure of the effective use of material of 'lightness' of the structure, and the minimization of the Daylight under the structure, a measure of the effective 'functionality' of the structure, are two quantifiable objectives. Thus, they can be easily put up when using any MOEA to optimize the system. A trade-off between these objectives can be evidenced trough the generation of the Pareto front after optimization. The difficulty here concerns only with the interfaces between the software's used, i.e., the optimization routine (developed in house) and the 3D design and building analysis software (commercial softwares) used to calculate the objective values. Since these commercial software's do not run in background a specific interface approach based on Windows operating system scripts was implemented. This script simulates the use of the programs used.

Since the third objective is not quantifiable and, additionally is very subjective, a different strategy was adopted which takes into account the preferences of the DM involved. This can be seen as an iterative process: i) first the MOEA generates the Pareto fronts using the Area and Daylight objectives; ii) then, the DM selects the preferred regions taking into account aesthetics; iii) this information is inserted on the MOEA and new optimization is carried out. The process is repeated until a satisfactory solution is found by the DM.

Therefore, the resolution of this type of problems involves the articulation of preferences of a DM. In the present case the selection made by the DM, concerning one or more regions of the Pareto front, implies the definition of a measure of the relative importance of the objectives considered (in the present case two objectives exist, Area and Daylight). This can be better illustrated with the example of Figure 2. In region 1 the Area has more importance, since these solutions have better value for the Area, while in region 2 the Daylight is the most important objective (both objectives are to be minimized).

A traditional way to deal with multi-objectives consists in using an aggregation function, such as the weighted sum, were the relative importance of the various objectives are taking into account trough the definition of weights [14]. In general terms three different classes of multi-objective preference methods can be identified, depending on how the search and decision processes are interconnected, a priori, a posteriori and iterative methods [15,16]. In a priori methods, the DM must specify her or his preferences, expectations and/or options before the optimization process takes place. The preferences are expressed in terms of an aggregating function which combines individual criterion values into a single utility value. In the case of a posteriori method, after the generation of the Pareto optimal set, the DM selects the most preferred among the alternatives

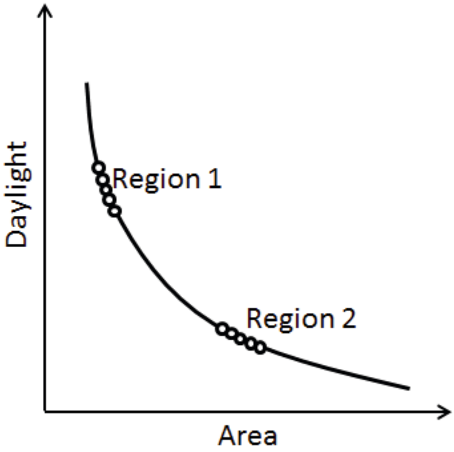


Fig. 2. Trade-off between Area and Daylight

taking into account his or her own preferences. Finally, in interactive methods the Decision making and the optimization processes occur at interleaved steps. At each step, partial preference information is supplied by the DM to the optimizer, which, in turn, generates better alternatives according to the information received.

Therefore, in the iterative methodology proposed different preferences methods are used (see Figure 3). At the beginning the MOEA runs without considering any preference and considering only the quantifiable objectives. After the Pareto front is generated, the DM selects the preferred region based on aesthetics parameters. The major difficulty consists in incorporating the information concerning the regions selected on the MOEA. The idea is to use a priori decision making methodology proposed before, which is based on the use of stress functions [17]. In this method the incorporation of preferences is made through the definition of a set of weights quantifying the relative importance of the objectives. The value calculated for the stress function depend on the objective function itself as well of the weight chosen for this objective. The extension of the Pareto front found depend on the definition by the user of an algorithm parameter. For more details the reader is referred to [17]. Starting from a population of solutions resulting from the previous optimization run the algorithm searches for solutions in the region corresponding to the weights chosen. However, care must be taken since the usability of interactive methods depends strongly on the extent to which the parameter values set by the DM as an expression of his or her preferences lead to solutions corresponding to those preferences.

Another important issue concerns the huge search space, which is a characteristic of this type of design problems (as will be seen on the problem tested below). In this case some of the solutions found, which are valid when calculating the Area and Light objectives, have some risk of not being valid concerning

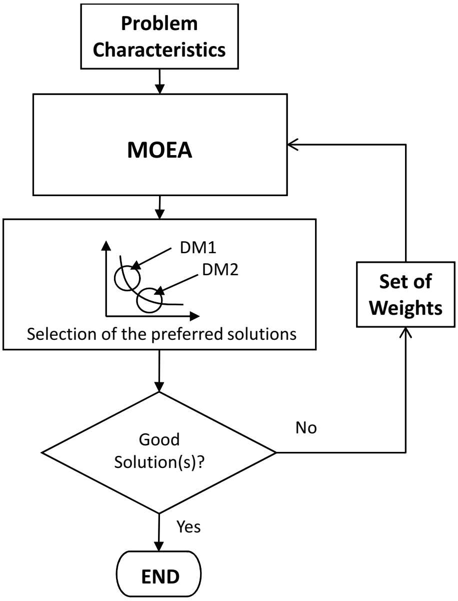


Fig. 3. Combination of a posteriori and iterative methods to select solutions in multi-objective problems involving aesthetic design variables

other questions such as possibility of fabrication. This aspect will not be deal in this phase of the work. However, a decision must be taken about the boundary conditions imposed to the decision variables. If the range of variation allowed is high the Pareto front obtained will have, certainly, solutions with very different aesthetics. If the range of variation is very restrictive, the possibility of losing some important designs (solutions) is high.

4 Optimization Strategy

In this section the strategy proposed to deal with the problems identified above, i.e., multiple objectives, non-quantifiable objectives and size of the search space, will be described in detail. The resolution of this type of problems can be made using three different situations:

Situation 1: The simplest situation consists in using the optimization algorithm (MOEA) without interacting with the DM (i.e., only one time). The DM defines the decision variables to be optimized and their range of variation and the objectives to be considered. Then, after running the MOEA, the DM selects the solutions from the pool of non-dominated solutions obtained using, for example, aesthetics criteria. In this case the DM must know very well the characteristics of the problem to be solved, since it is necessary to define beforehand the boundary conditions imposed to the design variables.

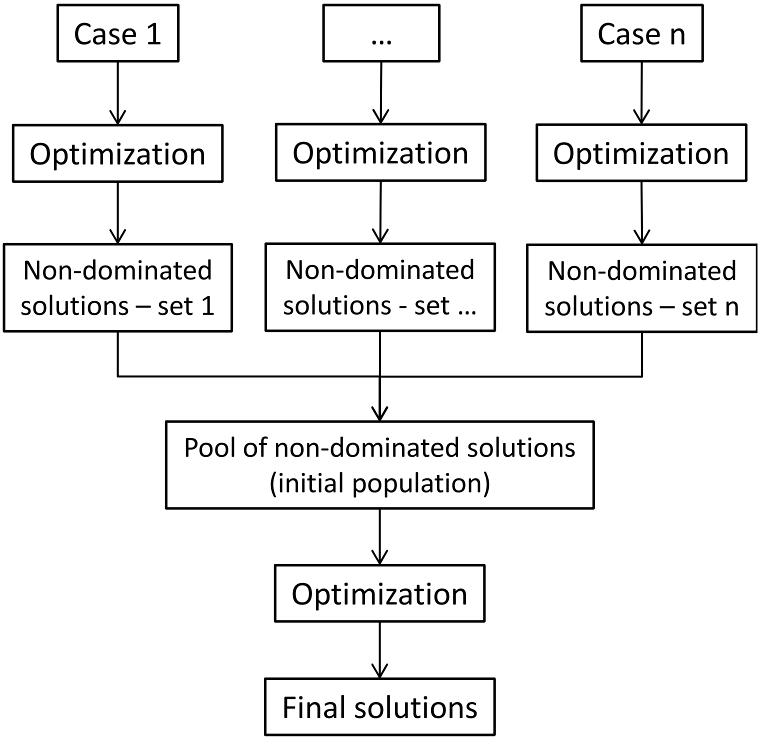

Case 1
...
Case n
Optimization
Optimization
Optimization
Non-dominated solutions – set 1
Non-dominated solutions - set ...
Non-dominated solutions – set n
Pool of non-dominated solutions (initial population)
Optimization
Final solutions

Fig. 4. Global structure of the optimization strategy adopted

Situation 2: This situation is illustrated in Figure 3. In this case an iterative process is pursued. In each interaction, new information (e.e., a set of weights) as provided by the DM, is taken into account. However, as in previous situation, the DM has to define the decision variables to be optimized and their range of variation and the objectives to be considered. Thus, the results produced will be certainly strongly dependent on the initial choice made by the DM.

Situation 3: This situation is illustrated in Figure 4. It starts by the definition of n different cases, each one characterized by different set of restrictions (i.e., boundary conditions) imposed by the DM to the decision variables. Then, each one of these cases is optimized independently. At the end of this initial optimization step, the best solutions selected from all n cases will be used to form a new population of solutions. This population will serve as initial population for the last optimization process. The optimization step in this case can be performed either using a simple MOEA optimization (as in situation 1) or using an iterative process (as in situation 2). It is expected that the non-dominated solutions found have characteristics taken from the different cases (i.e., different set of restrictions imposed).

5 Example of Application

5.1 Problem to Solve

Different geometrical boundary conditions are input by the user, in order to explore different conceptual solutions. In the present work, 3 different geometrical boundary conditions (i.e., 3 different cases as represented in Figure 4) were used, each one leading to a different optimized subset of solutions. The surface was defined by 20 control points and defined by the NURBS method (see Figure 1). The natural light levels are calculated in Ecotect [13] over a horizontal analysis grid at ground level. The grid was formatted with a dimension of 5x5 meters and was set to a 5x6 matrix allowing for calculation over all 30 visible nodes. Calculations of natural light levels are neither time nor date dependant, so no parameters were specified and the default values of the software were used.

In case 1, the less restrictive, the coordinates of the 20 control points represented in 1 (corresponding to 60 decision variables, the 3D coordinates of the control points) are allowed to vary between 0.5 and 5 meters. In case 2 the control points corresponding to the corners of the structure are fixed, i.e., points P1(0,0,0), P4(5,0,0), P17(0,5,0) and P20(5,5,0). In this case 48 decision variables are to be optimized. Finally, in the most restrictive case (case 3), the coordinates of the control points corresponding to the corners points as well to the border points are fixed, i.e., points P1(0,0,0), P2(1.6,0,0.5), P3(0.338,0,0.5), P4(5,0,0), P8(5,0.65,0.18), P13(0,0.335,0.18), P16(5,0.335,0.18), P17(0,5,0), P18(1.6,5.0,0.5), P19(0.338,5,0.5) and P20(5,5,0). This corresponds to 24 decision variables. In cases 2 and 3 the coordinates of the remaining control points are allowed to range in the interval [0.5, 5] meters (as in case 1).

After this process the user is presented with the geometrical solutions and their performance, and allowed to bias the subsequent optimization step towards his/her preference (assumed to be based on the aesthetics of the solutions provided). The solutions selected are used as initial population for the final optimization. In this case no restriction to the decision variables are imposed, thus 60 decision variables are considered. They are allowed to range in the interval [0, 5] meters, the aim being to cover all possible solutions generated in the previous optimization cases.

The MOEA adopted in this work is the Reduced Pareto Set Genetic Algorithm (RPSGA) proposed before by one of the authors [18,19]. The values of the parameters inside the RPSGA are the best values as described in [19]. The main and elitist populations had 100 and 200 individuals, respectively; a roulette wheel selection strategy was adopted; a crossover probability of 0.8, a mutation probability of 0.05, a number of ranks of 30 and limits of indifference of the clustering technique of 0.01 were chosen. In all cases the algorithm ran only during 10 generations due to the computation time required by the modeling software.

5.2 Optimization Results

Figures 5 to 7 shows the initial population and the non-dominated solutions of the 10th generation, as well 3 different optimized designs of the roof structure, for

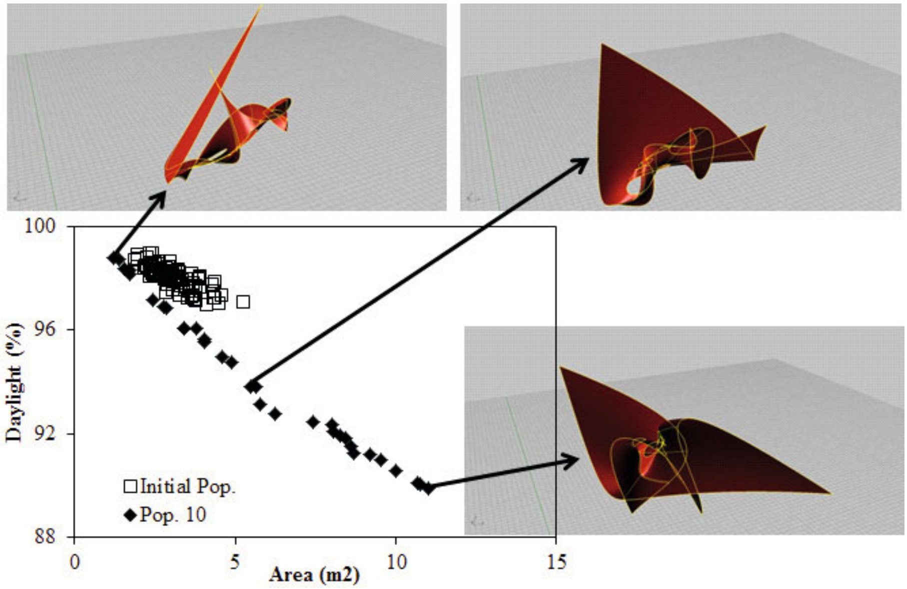


Fig. 5. Pareto frontier for case 1

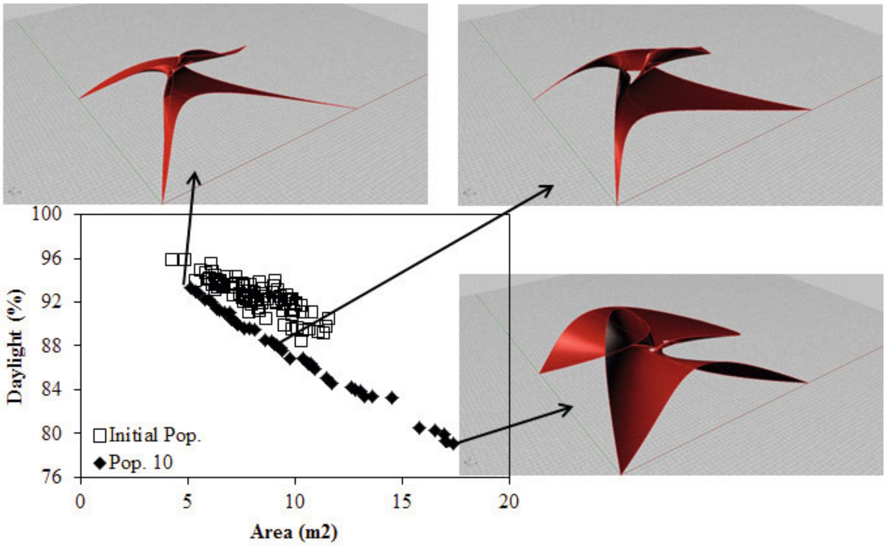


Fig. 6. Pareto frontier for case 2

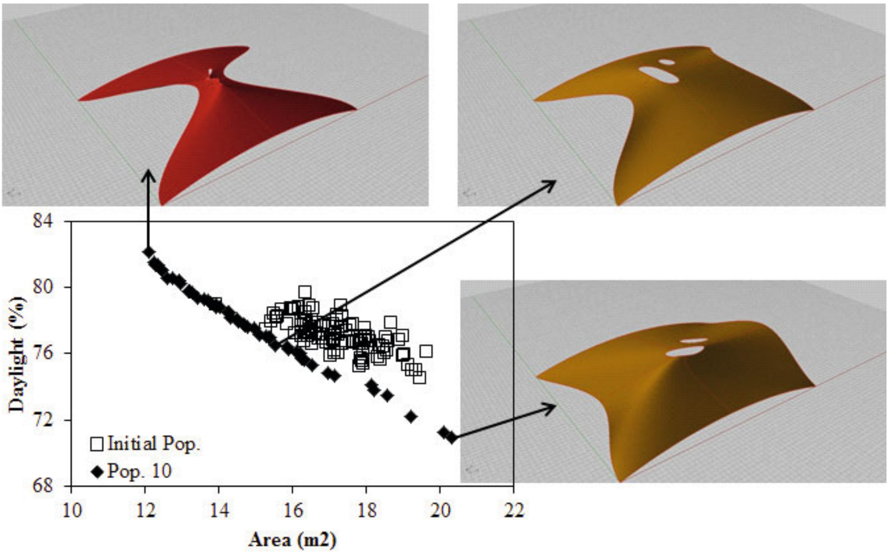


Fig. 7. Pareto frontier for case 3

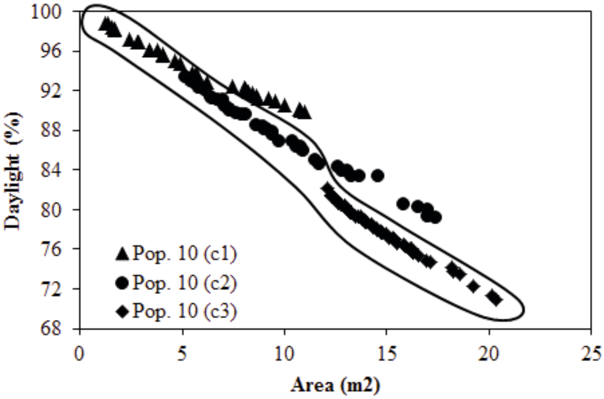


Fig. 8. Initial population for final optimization (non-dominated solutions of cases 1 to 3)

cases 1 to 3, respectively. As can be seen the algorithm is able to evolve during the 10 generations and the Pareto frontier obtained in each case is well distributed. As expected, the roof structures obtained in case 1 are very random, while in the other two cases the structures obtained are coherent with the boundary conditions defined. In case 2 the corners are well defined and in case 3 this is also true for the four sides of the structure.

From the Pareto solutions of these three cases a new Pareto front was defined as illustrated in Figure 8. This set of solutions was the initial population of

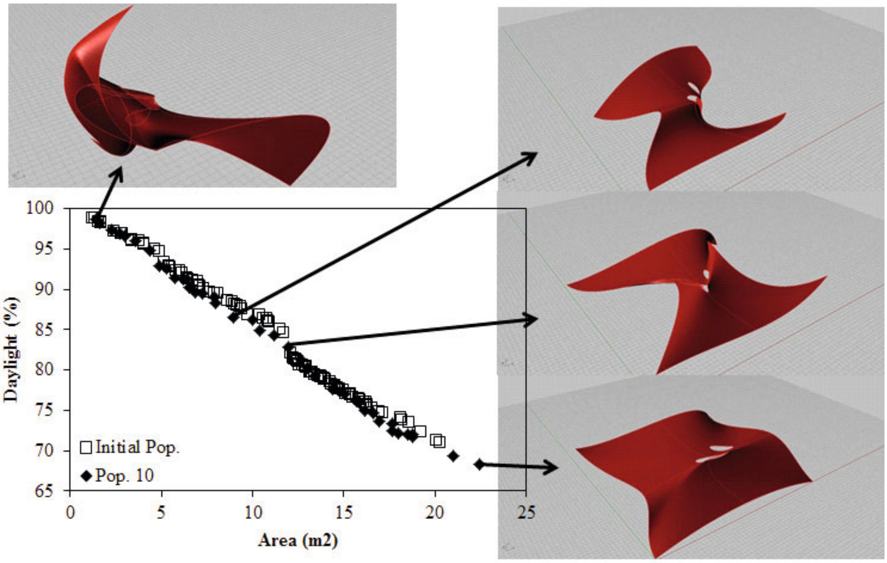


Fig. 9. Global optimization: initial population and non-dominated solutions after 10 generations

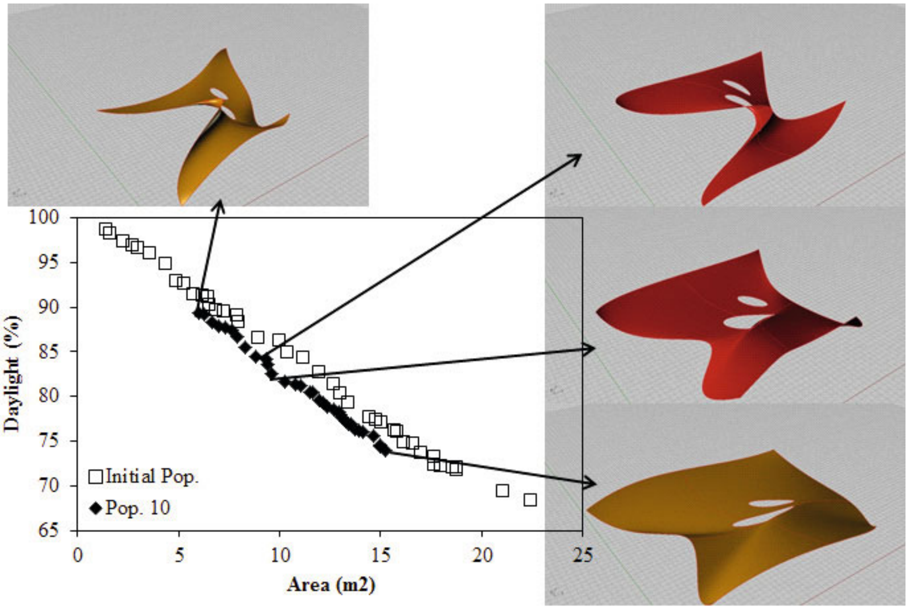


Fig. 10. Optimization considering a weight vector of (0.5; 0.5) and as initial population the population resulting from previous run (Figure 9)

the last optimization process, as identified in global strategy adopted in this work (Figure 4). It is interesting to note that the Pareto solutions of the three previous runs were able to define almost a continuous line. As can be observed in the optimization results presented in Figure 9 the MOEA was able to fill some of the gaps between the solutions of the initial population and some improvements are obtained. As expected, the new non-dominated solutions are able to cover all type of geometries obtained in the first three runs. Now the DM has the starting point for selecting the preferred geometries having a very good idea of their performance in terms of the two quantifiable objectives defined initially. This is done in Figure 10 where a set of weights of (0.5, 0.5) was selected by the DM, assuming that he/she "likes" the designs present in the center of the Pareto front (Figure 9). The decision making methodology based on the stress function is applied using as initial population the population found before. This method was able to obtain much better solutions than the previous ones and, simultaneously, converge for the preferred region of the Pareto front. At this point the DM can continues the process by selecting a new set of weights and/or by imposing additional restriction on the size of the portion of the Pareto front to be obtained.

Finally, it is important to note that in the generic roof structure two “skylights” were designed. Those “skylights”, besides providing more light under the structure, are a fundamental design characteristic of this object, and contribute to the aesthetic perception of the geometry. But formal characteristics of a surface can change dramatically under the manipulation of the control points. As a result many of the intermediate solutions, although performing much better on the quantitative evaluation criteria, will not classify as aesthetically pleasing and will therefore be discarded or excluded by the DM from the next pool of solutions.

6 Conclusions

Design is about decision making and requires judgment and trade-offs based on the best available information. Therefore the role of optimization in design is to provide the designer with quantitative and qualitative information. This information is a way for increasing the designers understanding of the design problem and the nature of good solutions.

Design decisions made in the early stages of the design process have a higher effect on the final performance and outcome compared to decisions taken at later stages of the design process. Therefore the strategies which are followed in the beginning of a design project and the decisions made during those early stages are most important. Generative systems are an essential part of the future development of performative architectural systems where evolutionary principals are applied in the initial stages of the design process with the intent to automate explorative research. The outcome of those processes is expected to be surprising and inspiring.

This study has introduced the use of a MOEA in the conceptual phase of the design process. The applied strategy for the use of a MOEA allowed for the DM to iteratively control the outcome and steer the process to a personal aesthetical solution. The DM can rely less on intuition to solve complicated and conflicting design requirements and concentrate efforts on innovative and aesthetical pleasing results.

The next step in this research is to demonstrate the applied design method and this specific MOEA for the design of an architectural object which can be tested and validated in the real physical world. In addition the method could be further developed and prepared for general use by less computer literate architects and designers for deployment in real world design processes.

Acknowledgements. One of the authors acknowledges the financial support received by the Portuguese Science Foundation under grant SFRH/BD/44600/2008.

References

1. Reffat, R.M.: Computing in architectural design: reflections and an approach to new generations of CAAD. The Journal of Information Technology in Construction 11, 655–668 (2006)
2. Oxman, R.: Theory and design in the first digital age. Design Studies 27, 229–265 (2006)
3. Kolarevic, B.: Architecture in the Digital Age: Design and Manufacturing. Spon Press, New York (2003)
4. Liu, Y.-T., Lim, C.-K.: New tectonics: a preliminary framework involving classic and digital thinking. Design Studies 27, 267–307 (2006)
5. Caldas, L.G.: An evolution-based generative design system : using adaptation to shape architectural form. PhD Thesis, Massachusetts Institute of Technology (August 23, 2005)
6. Shea, K., Aish, R., Gourtovaia, M.: Towards integrated performance-driven generative design tools. Automation in Construction 14, 253–264 (2005)
7. Kilian, A.: Design exploration through bidirectional modelling of constraints. PhD Thesis, Massachusetts Institute of Technology (August 25, 2006)
8. Kolarevic, B., Malkawi, A.M.: Performative Architecture: Beyond Instrumentality. Spon Press, London (2005)
9. van Hinte, E., Beukers, A.: Lightness: The Inevitable Renaissance of Minimum Energy Structures, 3rd edn., Uitgeverij, Rotterdam (1998)
10. Sass, L., Oxman, R.: Materializing design: the implications of rapid prototyping in digital design. Design Studies 27, 325–355 (2006)
11. Rogers, D.F.: An Introduction to NURBS with Historical Perspective. Morgan Kaufmann, London (2001)
12. Robert McNeel & Associates, www.rhino3d.com (accessed in 2010)
13. Autodesk - Ecotect, www.autodesk.com/ecotect (accessed in 2010)
14. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)

15. Miettinen, K.M.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
16. Hwang, C.L., Masud, A.S.: Multiple Objective Decision Making - Methods and Applications. Springer, Berlin (1979)
17. Ferreira, J.C., Fonseca, C.M., Gaspar-Cunha, A.: Assessing the quality of the relation between scalarizing function parameters and solutions in multiobjective optimization. In: IEEE Congress on Evolutionary Computation (IEEE CEC 2009), Trondheim, Norway (2009)
18. Gaspar-Cunha, A.: Modeling and Optimization of Single Screw Extrusion. PhD Thesis, University of Minho, Portugal (2000)
19. Gaspar-Cunha, A., Covas, J.A.: RPSGAe-A Multiobjective Genetic Algorithm with Elitism: Application to Polymer Extrusion. In: Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems. Springer, Heidelberg (2004)

Introducing Reference Point Using g-Dominance in Optimum Design Considering Uncertainties: An Application in Structural Engineering

David Greiner, Blas Galván, José M. Emperador, Máximo Méndez,
and Gabriel Winter

Institute of Intelligent Systems and Numerical Applications in Engineering (SIANI),
Universidad de Las Palmas de Gran Canaria, 35017, Spain
{dgreiner,jemperador}@iusiani.ulpgc.es,
{bgalvan,gabw}@step.es, maximo@dis.ulpgc.es

Abstract. Considering uncertainties in engineering optimum design is often a requirement. Here, the use of the deterministic optimum design as the reference point in g-dominance is proposed. The multiobjective optimum robust design in a structural engineering test case where uncertainties in the external loads are taken into account is proposed as application, where the simultaneous minimization of the constrained weight average and the standard deviation of the constraints violation are the objective functions. Results include a comparison between both non-dominated sorting genetic algorithm II (NSGA-II) and strength Pareto evolutionary algorithm (SPEA2), including S-metric (hypervolume) statistical comparisons with and without the g-dominance approach. The methodology is capable to provide robust optimum structural frame designs successfully.

Keywords: Engineering Design, Multiobjective Optimization, Structural Optimization, Frames, Steel Structures, Uncertainty, g-dominance.

1 Introduction

The inclusion of uncertainties in the problem variables are often required in engineering design, being these parameters not given fixed coefficients, but random variables. Concretely, in the field of structural design optimization, traditionally two approaches are considered: reliability-based design optimization (RBDO) and robust design optimization (RDO), (a recent reference is e.g. Tsompanakis et al. [18]).

Evolutionary multiobjective algorithms have been used in the RDO approach in computational solid, e.g. Lagaros et al. [14] or Greiner et al. [9] and computational fluid mechanics, e.g. Lee et al. [16], where the simultaneous minimization of the desired fitness function average (first objective) and its standard deviation (second objective) is considered. Also a multiobjective hybrid approach is used in Lagaros et al. [15] with a RBDO and a reliability-based robust design optimization (RRDO) in structural trusses.

The articulation of decision maker preferences can be expressed before (a priori), progressively during (interactive), and after (a posteriori) the optimization process. A posteriori, MOEA separate the optimization and decision making-process. First

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 389–403, 2011.

algorithms pursue to reach a set of candidate solutions and to sample it by an even distributed set of efficient solutions; then the decision maker chooses a solution according to its preferences. This approach is very popular in the MOEA community. In interactive methods, search processes and dialogs with the decision maker are alternated. At the end of this reiterated guided exploration process, the decision maker has a deep knowledge to adopt the most representative solution. In a priori MOEA the idea consists of incorporating the decision maker's preferences before the process of search and convergence to the preferred region of the Pareto front. In this context, a survey of representative work can be found in [1]. Recent advances in multicriteria decision making based evolutionary multiobjective algorithms have considered the use of a reference point in the optimization process: R-NSGA-II in Deb et al. [3] or g-dominance in Molina et al. [17]. In this paper, considering the reference point fitness values based in the deterministic optimum design (where the optimization is performed taken into account fixed parameters/variables without considering uncertainties) is suggested. The deterministic optimum design is often determined by the problem constraint limits. In case of including uncertainties, this deterministic design easily violates them, but constitutes a good reference where to evaluate the non-dominated front. Here, a multiobjective optimum robust design problem through a structural engineering test case where uncertainties in the external loads are taken into account is proposed as validation case. The simultaneous minimization of the constrained mass average and the standard deviation of the constraints violation are the objective functions, and g-dominance is used to fix the reference point.

The paper describes in the second section the structural problem including the deterministic approach and the uncertainty consideration approach, following with the proposal of considering the deterministic optimum design as reference point in g-dominance criterion when including uncertainties. The test case constitutes the fourth section, followed by results and discussion. Finally it ends with the conclusions, acknowledgment and references.

2 The Structural Problem

2.1 Deterministic Design

Bar structures designs have to be balanced between guaranteeing the fulfilment of its function (to maintain the structural loads: a) without collapsing or breaking -or even without surpassing some stress value frequently associated with yield stress or fatigue stress limits-; b) without suffering from unstable phenomena associated with the buckling effect; c) without surpassing certain structural displacements values which lead to unsafety) and doing it in an economic way. The former is roughly associated with increasing the bar cross-section types, while the latter is associated with diminishing them. Therefore the objective is to perform a constrained mass minimization where the fitness function has to consider these proper requirements of the bar structure to fulfil its function. Its value is directly related with the acquisition cost of raw material of the metallic frame. The information needed by the fitness function is obtained through a finite element code and the applied constraints in order to guarantee the appropriate functionality of the structure are defined in terms of stresses, compressive slenderness and displacements:

a) Stresses of the bars, where the limit stress ($\sigma_{\lim}$) depends on the frame material and the comparing stress (σ_{co}) takes into account the axial and shearing stresses by the shear effort, and also the bending effort. For each bar, Equation (1) has to be accomplished:

$$\sigma_{com} - \sigma_{\lim} \leq 0, \tag{1}$$

being σ_{co} the comparing stress of each bar, and $\sigma_{\lim}$ the stress limit.

b) Compressive slenderness limit, for each bar where the buckling effect is considered (depending on the code used it could have different values) Equation (2) has to be satisfied:

$$\lambda_{com} - \lambda_{\lim} \leq 0 \ , \tag{2}$$

being λ_{co} the slenderness of each bar, and $\lambda_{\lim}$ the compressive slenderness limit.

c) Displacements of joints or middle points of bars are also a possible requirement, as observed in Equation (3), valid for each displacement constraint:

$$u_{com} - u_{\lim} \leq 0, \tag{3}$$

being u_{co} the comparing displacement of each required joint or point, and $u_{\lim}$ the displacement limit (as in equations (1) to (3), the constraints limits –lim– are predefined parameters and the comparison magnitudes –com– are variables obtained from the finite element analysis).

With these constraints, the fitness function *constrained mass*, which integrates the constraints violations as mass penalties, is shown in Equation (4).

$$FitnessFunction = \left[\sum_{i=1}^{Nbars} A_i \cdot \rho_i \cdot l_i \right] [1 + k \cdot \sum_{j=1}^{Nviols} (viol_j - 1)] \tag{4}$$

Where:

A_i = area of the section type of bar i; ρ_i = density of bar i; l_i = length of bar i; k = constant that regulates the equivalence between mass and restriction (suitable values around the unity order); $viol_j$ = for each violated restriction j, is the quotient between the violated restriction value (stress, displacement or slenderness) and its reference limit.

No uncertainties (that is, no random variables consideration) are taken into account in the previously exposed deterministic design problem, which is defined as a mono-objective optimum engineering design problem (mass minimization) and has been solved extensively in the literature (e.g. see [13]).

2.2 Design Including Uncertainties

The deterministic optimum design of a bar structure is defined frequently by the imposed constraints in terms of stress, displacement or buckling, which are conducted to their limit values, without surpassing them. The variation condition in loads is in real structures frequent, and it is considered in the design codes. So, a deterministic optimized structure, due to the fact that has their constraints near the limit values is expected to be more sensitive to those random variations. An analysis of those uncertainties is required to guarantee a robust design. The principal objective of robust design is to find a solution with less sensitive structural performance to the fluctuations of variables and/or parameters without eliminating their variation. Here, uncertainties in term of the variation of external parameters, the structural loads, are considered. The variation of the load actions which act over a structure from the viewpoint of the probabilistic or semi-probabilistic safety criteria, is associated with considering the loads as stochastic variables and to the existence of some: a) limit ultimate states that guide to the total or partial ruin of the structure and, b) limit service states that when achieved produce its malfunctioning. Here, in order to define the actions, it is assumed that their variation follows a Gaussian probability density function. The characteristic value of an action is defined as the value that belongs to the 95% percentile, that is, a probability of 0.05 to be surpassed, which is considered as the deterministic value in no uncertainty consideration case. In this paper, in order to model the stochasticity of the actions, standard Monte Carlo simulations are performed considering variable distribution of the external loads.

The Monte Carlo methods are essentially based on the possibility of evaluating the integral I (Equation (5)) using the so called Horvitz-Thompson estimator $\hat{I}$ [10]. To do so, a set of independent random samples $X_1,X_2,\ldots,X_n$ drawn according to a probability density function $p(x)$ are used. If the domain is $\Omega=[0,1]^s$, the samples X_i are independent and uniformly distributed, then $\hat{I}$ yields the correct result on average.

$$I=\int_\Omega f(x)d\mu(x) \qquad \hat{I}=\frac{1}{N}\sum_{i=1}^{N} f(X_i) \qquad V\left[\hat{I}\right]=\frac{1}{N}\left[\int_\Omega \frac{f^2(x)}{p(x)}d\mu(x)-I^2\right] \tag{5}$$

The error made in the approximation is, on average, $\sigma(f)/\sqrt{N}$ being the approximation $O(N^{1/2})$. Monte Carlo converges in spaces of any dimension, without taking into account the regularity or smoothness of the integrand, but the efficiency of the simulation can be enhanced using some Variance Reduction [4], whose idea is to increase the efficiency of the simulation process using known information about the system being simulated. A high number of variance reduction techniques have been developed during the past decades, for many and very different situations, among them the so called “Importance Sampling” and those based on Markov Chain Monte Carlo techniques (e.g. Gibbs Sampling and Metropolis-Hastings algorithms) are very efficient for many problems. Monte Carlo has been utilized, among others, to simulate the large complex system models in which the usage of deterministic methods is very costly or impossible.

The exposed method is also applicable in case of uncertain values of material properties (such as Young Modulus) or uncertain values of the loads directions, but it has not been considered here.

3 Deterministic Optimum Design as Reference Point for Robust Optimization

When no uncertainties are considered, the optimum design is called deterministic optimum design, and, in case that constraints are imposed, this deterministic optimum design should satisfy all of them; that is, no constraint violations are allowed in this design. Considering the same problem, but now with some of the external variables/parameters as random variables, we are dealing with uncertainty. As the previous deterministic optimum design appears frequently as being in the border of the constraints violation, when considering this external random variables, is highly probable that the deterministic optimum design violates the constraints in this new uncertain environment.

A robust design achieves designs with good performances and simultaneously, is minimum sensitive to external variations (random variables). In our structural frame optimization problem, that means to achieve designs with low constrained mass average (good performance) and low constraints violations standard deviations (minimum sensitive to external variations). Both objectives are minimized. Therefore, the function values of the deterministic optimum design are good estimators of a lower limit of the solutions when considering uncertainties in the optimization (it is expected that those deterministic optimum values dominate the designs of the problem when considering random variables). That is, it is a good approach to consider those values as the reference point in the g-dominance criterion. G-dominance was introduced in [17]. It produces a reduced set of non-dominated solutions adapted to the decision maker preferences when they are expressed in terms of a reference point (aspiration level containing desirable values for the objective functions). The search space is divided in regions according to this reference point. G-dominance enhances those solutions: a) that either dominate the reference point or b) which are dominated by the reference point, discriminating all the others. So, when we are interested in the environment of a certain point in the objective space, it is a very useful tool. From the perspective of decision making, the preference information of the decision maker is given here as the reference point coordinates. Two ways of implementing g-dominance are proposed in [17]: a) changing the dominance-checking function and, b) changing the evaluation of the objective functions by greatly penalizing those solutions which do not belong to the area of interest as described previously. Here the latter (also claimed by the g-dominance authors as the simplest way to implement g-dominance) is used both in NSGA-II (Deb et al. [2]) and SPEA2 (Zitzler et al. [21]).

So, the deterministic optimum design objective function values are taken as the reference point in g-dominance for multiobjective optimization when considering uncertainties in the design.

4 Test Case

The considered reference test case is based on a problem taken from Hernández Ibáñez [11] for single objective mass minimization using continuous variables. The solution reported in the previous reference using classical optimization methods was improved using evolutionary algorithms in Greiner et al. [6]. This last deterministic evolutionary optimum design is taken as reference in this work and compared with the robust optimum design non-dominated front.

Figure 1 shows the test case, where lengths (10 and 20) and height (6) are in meters and the loads in T/m. (1.5, 1.0 and 0.2). There is a constraint of maximum displacement of middle point of bar 2 equal to length/300, that is 6.67 cm. It is a discrete domain problem, belonging the cross-section types to the IPE class (16 different types per bar). It has been taken into account the buckling effect, and also its own gravitational load. The considered density (7.85 T/m^3) and Young modulus (2100 T/cm^2) are common steel values and the yield stress is 235.2 Mpa.

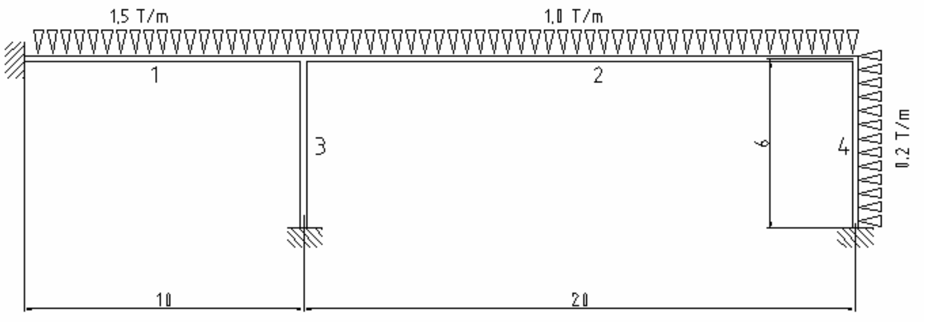


Fig. 1. Structural Test Case

The Monte Carlo simulation has been performed considering 30^N simulations per structural design in order to construct its constraint violation distribution, being N the number of different variables considered. Here the simulated variables correspond to the linear uniform loads of the frame structure which are three, belonging to each loaded bar (1, 2 and 4).

The distribution of each linear uniform load is simulated through a Gaussian distribution, which is calculated considering the test case load value as the characteristic value and its coefficient of variation being 6,1% for the vertical loads (bars 1 and 2) and 30,5% for the lateral load (bar 4). Their distributions are graphically represented in Figure 2.

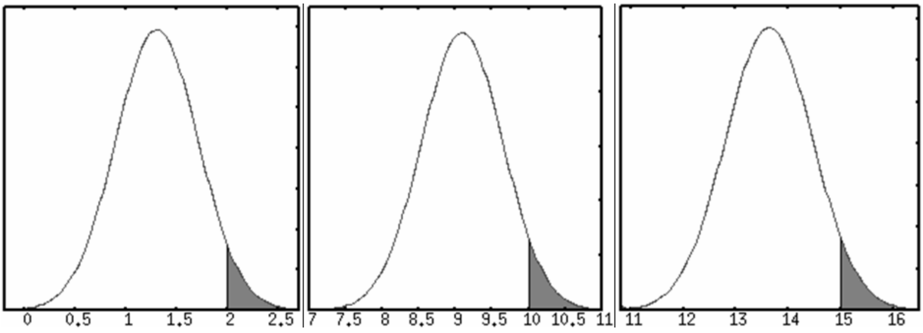


Fig. 2. External Load Distributions (Gaussian); Bars 4, 2 and 1, respectively

5 Results and Discussion

Ten independent runs of the evolutionary optimization design were executed in each case: NSGA-II, and SPEA2, both with implemented g-dominance relation. A population size of 200 individuals and 6% mutation rate (following results obtained in [7]) were used with Gray coding as described in Whitley et al. [19] (it has been found to be advantageous compared to standard binary in discrete frame structural design in [8]) with uniform crossover.

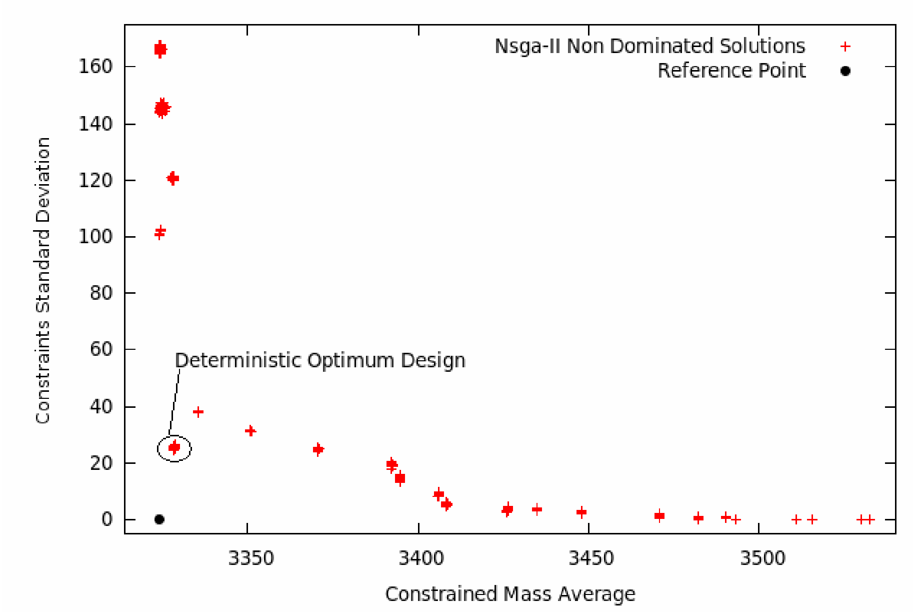


Fig. 3. G-dominance with deterministic design fitness function values as reference point. Non-dominated final optimum front function evaluations of the ten independent executions. NSGAII

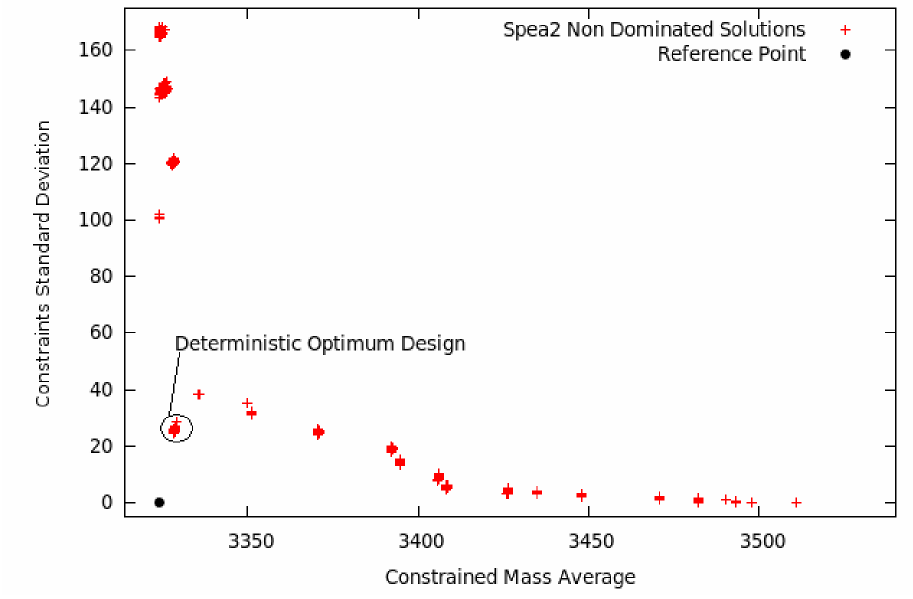


Fig. 4. G-dominance with deterministic design fitness function values as reference point. Non-dominated final optimum front function evaluations of the ten independent executions. SPEA2

The reference point was set from the value of the deterministic optimum design (taken from [6]): a mass of 3324.3 kg. for the x-coordinate and 0.0 kg. for the y-coordinate, as there are no constraints violation in the deterministic optimum when no uncertainties are considered. This reference point is depicted as a black point in figures 3 and 4. Dealing with robust design, when uncertainties are included, the optimum structural designs are expected to weight more due to the penalty function strategy (as described in section 2.1) because of the possible constraint violation; as well as the constraints violation standard deviation is expected to increase because of the same reason. Therefore, this deterministic optimum solution is expected to dominate the robust optimum design solutions, being also their objective function values desirable to be achieved. So, it is a good selection of the reference point in the g-dominance.

In figures 3 and 4, the non-dominated solutions of the final fronts belonging to each of the ten independent cases of NSGA-II and SPEA2 are shown, respectively. As some of the cases do not achieve the best non-dominated solutions, there are some points dominated by others in these figures. Some detailed solutions are represented in tables 1 and 2: the lowest mass design has been selected from each of the non-dominated solutions of each algorithm. Both achieve the same thirteen designs, being the deterministic optimum design (third line in bold in both tables) also included as member of the best robust structural designs set. Their representations in the functional space are highlighted in figures 3 and 4.

According to the results, there are ten structural designs with lower constraint violation, having the one with 3492.0 kg. the lowest structural mass. Even considering uncertainties, this last design allows a null constraint violation. So, beginning with the structural design that corresponds to the deterministic optimum, there are eleven choices from lower to higher robustness until a complete absence of constraints violation. Therefore, the suggested approach of considering the reference point of g-dominance as the values of the objectives function corresponding to the deterministic design without uncertainties considerations when optimizing with uncertainties has succeed. Even more, its design in the robust optimization is located the nearest in the functional space -Figs 3 and 4- (the deterministic optimum design has two different materializations in terms of objective functional space in these figures: a) without considering uncertainties, it constitutes the g-dominance reference point, or black point in the figures; b) considering uncertainties, it constitutes the surrounded red points labeled as deterministic optimum design in the figures).

In tables 1 and 2, some detailed designs are shown (the one with lower constrained mass of each design has been selected). The cross section types are identical in both cases for the thirteen structural designs. The differences in numerical values of the third first columns between NSGA-II and SPEA2 are due to the stochastic nature of Monte Carlo simulations; in any case these differences are minor and unimportant.

To compare the outcome of the whole front during the optimization process of the g-dominance approach versus the standard approach, we will evaluate the S-metric (hypervolume, originally proposed by Zitzler and Thiele [20]), which is a distinguished unary quality measure for solution sets in Pareto optimization [12], taking into account both coverage and approximation to the optimum front. Concretely, we use the S-metric proposal of Fonseca et al. [5] (Source code available at: http://sbe.napier.ac.uk/~manuel/hypervolume). The reference point considered is ($3^{\cdot}10^4$, 2000), whose values are chosen in order to guarantee that they are dominated by the candidate designs generated by the evolutionary multiobjective algorithms. Values of the S-metric in figures 5 to 10 have been normalized using as lower and higher values the minimum and maximum values ($5.30567^{\cdot}\ 10^7$ and $5.33488^{\cdot}\ 10^7$, respectively) from the hypervolume averages and best sets among the ten independent cases of both evolutionary algorithms.

Table 1. Detailed Non-dominated Structural Designs belonging to NSGA-II executions (bold corresponds to deterministic optimum)

Constrained Mass (kg)	Standard Dev. of Constraints Violation (kg)	Average of Constraints Violation (kg)	Bar 1 Cross Section Type	Bar 2 Cross Section Type	Bar 3 Cross Section Type	Bar 4 Cross Section Type
3324	166.182	132.351	IPE360	IPE500	IPE300	IPE500
3324.04	100.624	47.607	IPE360	IPE500	IPE300	IPE550
3328.37	**25.144**	**4.055**	**IPE330**	**IPE500**	**IPE450**	**IPE500**
3370.5	24.228	3.791	IPE400	IPE550	IPE200	IPE450
3392.18	18.689	2.394	IPE400	IPE550	IPE220	IPE450
3394.54	13.402	1.299	IPE360	IPE550	IPE300	IPE450
3405.89	7.967	0.564	IPE330	IPE500	IPE500	IPE500
3408.38	4.984	0.225	IPE400	IPE550	IPE160	IPE500
3426.17	2.703	0.114	IPE400	IPE550	IPE180	IPE500
3447.77	1.901	0.05	IPE400	IPE550	IPE200	IPE500
3470.81	0.55	0.007	IPE400	IPE550	IPE220	IPE500
3482.26	0.126	0.001	IPE360	IPE550	IPE360	IPE450
3492.94	0	0	IPE400	IPE550	IPE160	IPE550

Table 2. Detailed Non-dominated Structural Designs belonging to SPEA2 executions (bold corresponds to deterministic optimum)

Constrained Mass (kg)	Standard Dev. of Constraints Violation (kg)	Average of Constraints Violation (kg)	Bar 1 Cross Section Type	Bar 2 Cross Section Type	Bar 3 Cross Section Type	Bar 4 Cross Section Type
3324	165.449	132.347	IPE360	IPE500	IPE300	IPE500
3324.01	100.658	47.573	IPE360	IPE500	IPE300	IPE550
3328.35	**24.551**	**4.031**	**IPE330**	**IPE500**	**IPE450**	**IPE500**
3370.55	23.974	3.839	IPE400	IPE550	IPE200	IPE450
3392.13	17.998	2.343	IPE400	IPE550	IPE220	IPE450
3394.52	13.086	1.276	IPE360	IPE550	IPE300	IPE450
3405.88	7.652	0.547	IPE330	IPE500	IPE500	IPE500
3408.38	4.767	0.221	IPE400	IPE550	IPE160	IPE500
3426.16	2.988	0.109	IPE400	IPE550	IPE180	IPE500
3447.77	2.029	0.051	IPE400	IPE550	IPE200	IPE500
3470.82	0.835	0.021	IPE400	IPE550	IPE220	IPE500
3482.26	0.157	0.002	IPE360	IPE550	IPE360	IPE450
3492.94	0	0	IPE400	IPE550	IPE160	IPE550

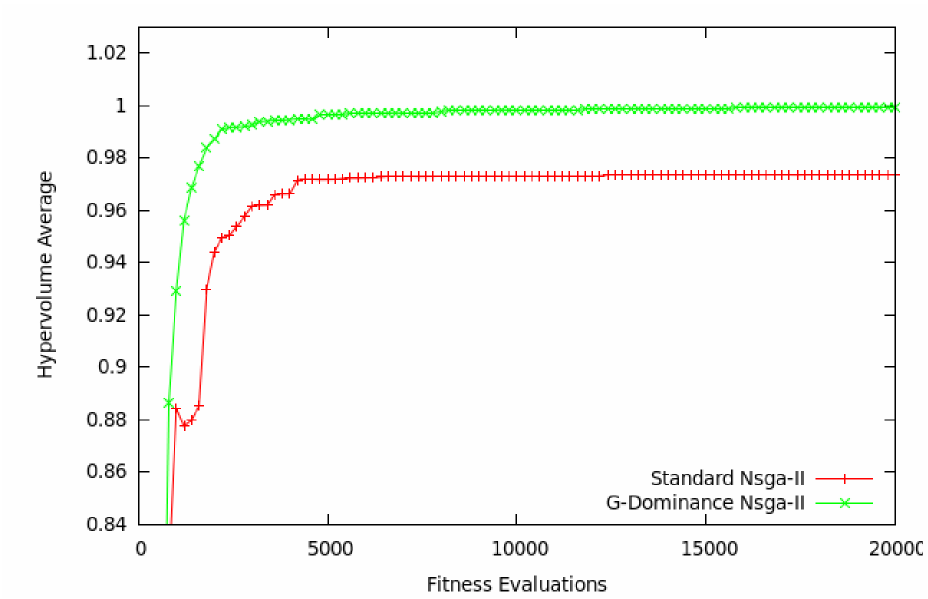


Fig. 5. Hypervolume average over ten independent executions comparing standard NSGA-II and NSGA-II with g-dominance approach

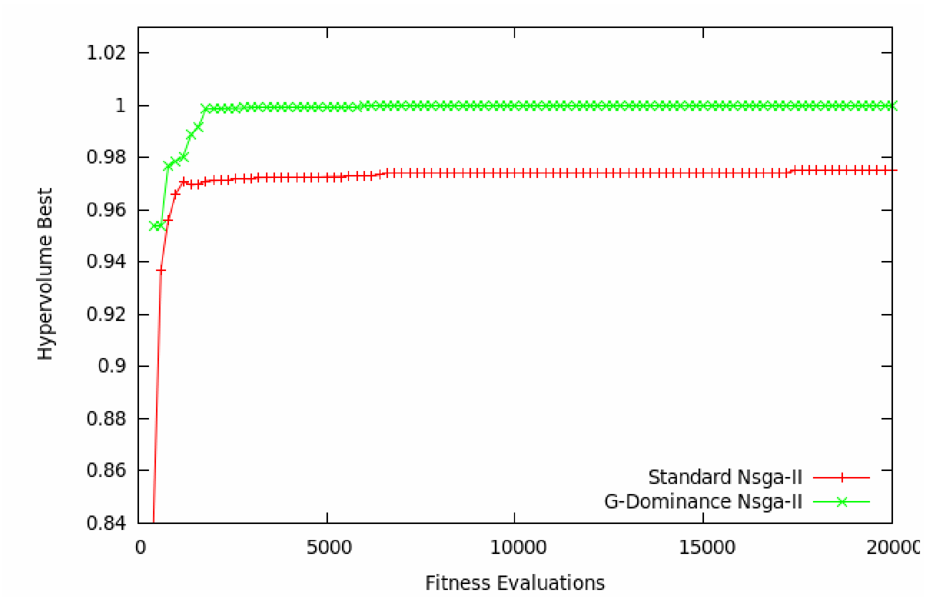


Fig. 6. Hypervolume best over ten independent executions comparing standard NSGA-II and NSGA-II with g-dominance approach

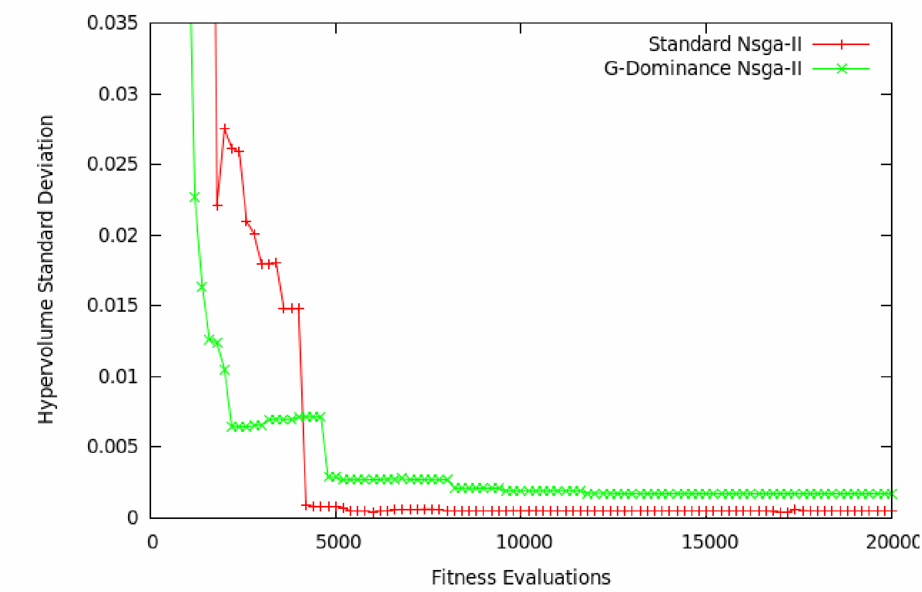


Fig. 7. Hypervolume standard deviation over ten independent executions comparing standard NSGA-II and NSGA-II with g-dominance approach

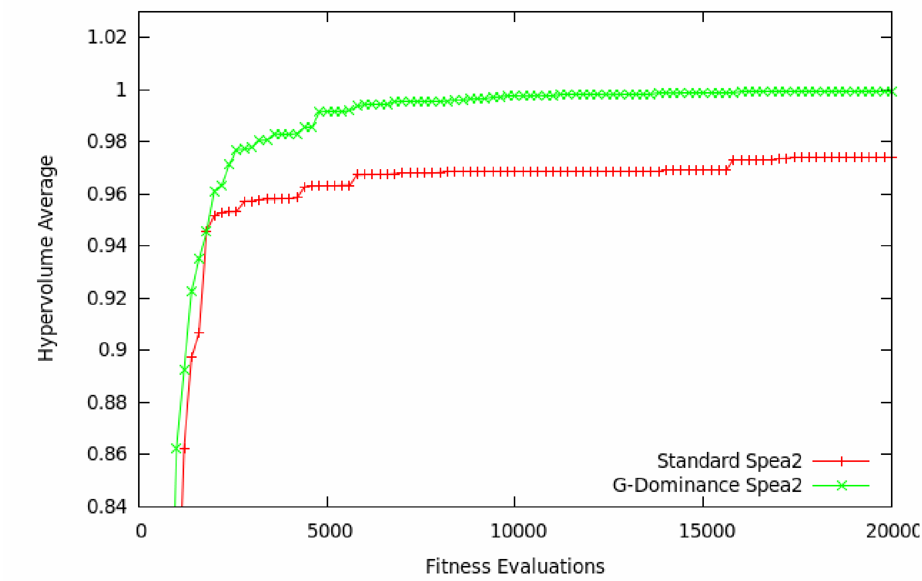


Fig. 8. Hypervolume average over ten independent executions comparing standard SPEA2 and SPEA2 with g-dominance approach

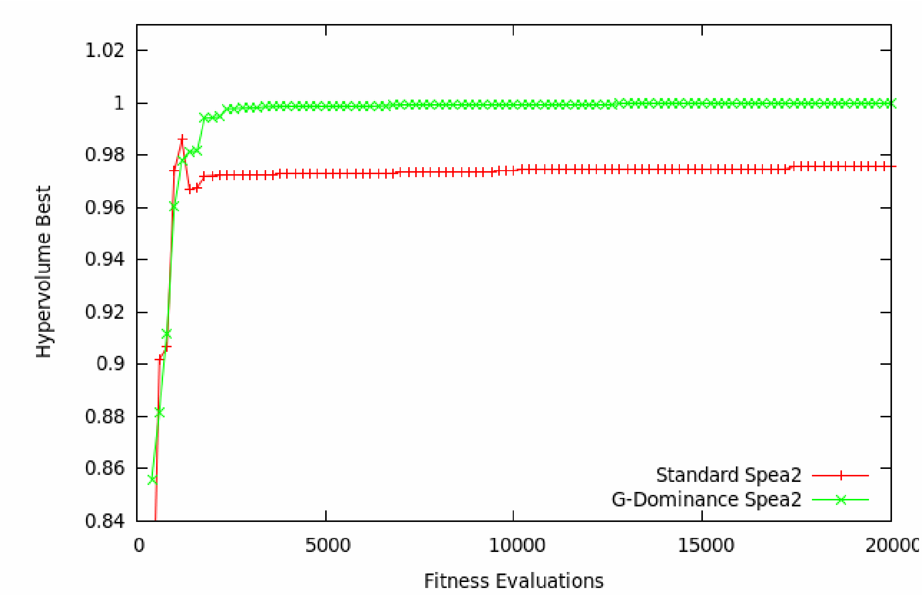


Fig. 9. Hypervolume best over ten independent executions comparing standard SPEA2 and SPEA2 with g-dominance approach

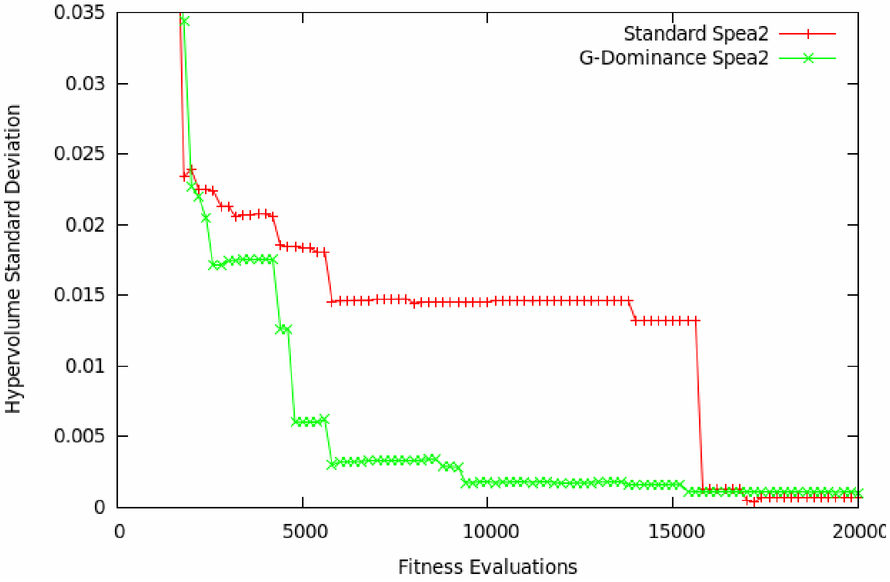


Fig. 10. Hypervolume standard deviation over ten independent executions comparing standard SPEA2 and SPEA2 with g-dominance approach

Results are graphically represented in Figures 5 to 10. The x-axis belongs to the number of function evaluations and the y-axis belongs to the hypervolume (S-metric) normalized values: average (Figs 5 and 8), best (Figs 6 and 9) and standard deviation (Figs 7 and 10) over the ten independent runs of NSGA-II (Figs 5-7) and SPEA2 (Figs 8-10) algorithms. The g-dominance approach hypervolume is compared versus the standard approach (conventional Pareto non-domination rule), being only the solutions dominated by the reference point (deterministic optimum solution values) considered for metric evaluation. Under this hypothesis, figures of average and best values over the ten independent executions show clearly the higher values achieved by the g-dominance approach, both in NSGA-II and SPEA2 algorithms. In the hypervolume average values this advantage is also present in the left part of the figure at the steepest section of the figure. Examining the hypervolume averages, the maximum average of standard NSGA-II (0.97371) achieved at 20,000 fitness evaluations is surpassed in case of g-dominance NSGA-II only with 1600 fitness evaluations (Fig. 5); also the maximum average of standard SPEA2 (0.97205) is surpassed in case of g-dominance SPEA2 only with 2600 fitness evaluations (Fig. 8). In the final stages of the evolution, the standard deviation of the g-dominance approach is slightly higher than the one of the standard approach.

In order to compare the behaviour of both evolutionary multiobjective algorithms when hybridized with the g-dominance approach, it is remarkable that the NSGA-II shows in this structural engineering robust optimum design application case a much faster convergence to the optimum non-dominated front than the SPEA2 at the

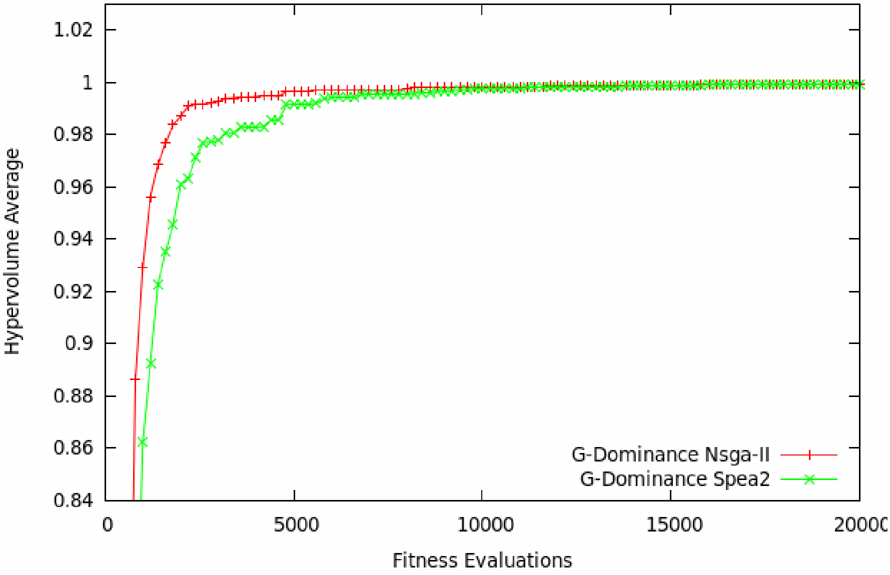


Fig. 11. Hypervolume average over ten independent executions comparing NSGA-II with g-dominance and SPEA2 with g-dominance approaches

earliest steps of the evolution. This is shown in Figure 11, where the average of the hypervolume over the ten independent executions is represented. This is also reflected in the hypervolume best values. Nevertheless, from the middle part of the evolution to the end, the obtained results of both algorithms are very similar, without significant differences.

6 Conclusions

A methodology using g-dominance has been proposed and implemented to use the deterministic optimum solution without uncertainties consideration, as reference point for the evolutionary multiobjective robust optimum design including uncertainties. As a first test case, a frame structure considering external loads as random Gaussian variables has been optimized here successfully with two evolutionary multiobjective algorithms (NSGA-II and SPEA2), showing the capabilities of the proposed methodology.

The choice of the reference point for g-dominance is pointed out to be key to contribute to the successful application of this novel methodology: the deterministic optimum design (whose genotype is known a priori) has been found the nearest to the g-dominance reference point in the objective space considering the robust optimization, and a set of close compromised optimum designs have been obtained as candidates to solve efficiently the structural engineering problem with uncertainties consideration, (which includes the solution with no constraints violation).

In the future it would be interesting to extend this optimum design methodology to other structural problems and other fields of engineering problems (e.g.: aeronautical, safety systems design), where the consideration of uncertainties in the optimization process is undoubtedly of value, and the integration of multicriteria decision making tools such as the g-dominance procedure introduced here could facilitate to solve them.

Acknowledgement

This work has been partially funded by the research project ULPGC2008-009. Also computational resources are supported by the project UNLP08-3E-2010 of the Secretaría de Estado de Universidades e Investigación, Ministerio de Ciencia e Innovación (Spain) and FEDER. The authors thank the anonymous reviewers for their helpful and constructive comments.

References

1. Branke, J.: Consideration of Partial User Preferences in Evolutionary Multiobjective Optimization. In: Branke, J., et al. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 157–178. Springer, Heidelberg (2008)
2. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm NSGAII. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
3. Deb, K., Sundar, J., Bhaskara, U., Chaudhuri, S.: Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. International Journal of Computational Intelligence Research 2(3), 273–286 (2006)
4. Fishman, G.S.: Monte Carlo: Concepts, Algorithms and Applications. Springer Series in Operations Research. Springer, New York (1996)

5. Fonseca, C., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. IEEE Congress on Evolutionary Computation, 1157–1163 (2006)
6. Greiner, D., Winter, G., Emperador, J.M.: Optimising Frame Structures by different strategies of genetic algorithms. Fin Elem Anal. Des. 37(5), 381–402 (2001)
7. Greiner, D., Winter, G., Emperador, J.M.: A comparative study about the mutation rate in multiobjective frame structural optimization using evolutionary algorithms. In: Schilling, R., et al. (eds.) Proceedings of the Sixth Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems –EUROGEN–, Munich, Germany (2005)
8. Greiner, D., Winter, G., Emperador, J.M., Galván, B.: Gray Coding in Evolutionary Multicriteria Optimization: Application in Frame Structural Optimum Design. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 576–591. Springer, Heidelberg (2005)
9. Greiner, D., Emperador, J.M., Galván, B., Winter, G.: Robust optimum design of metallic frames under uncertain loads. In: Congress of Numerical Methods in Engineering, Barcelona, Spain (2009)
10. Hammersley, J.M., Handscomb, D.C.: Monte Carlo Methods. Chapman and Hall, New York (1964)
11. Hernández Ibáñez S.: Structural Optimal Design Methods. Seinor Collection, Col. Ingenieros Caminos, Canales y Puertos, Madrid (1990)
12. Knowles, JD., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich (February 2006)
13. Kicinger, R., Arciszewski, T., De Jong, K.A.: structural design: A survey of the state of the art. Computers and Structures 83, 1943–1978 (2005)
14. Lagaros, N.D., Papadrakakis, M.: Robust seismic design optimization of steel structures. Struct. Multidisc. Optim. 37, 457–469 (2007)
15. Lagaros, N.D., Tsompanakis, Y., Fragiadakis, M., Plevris, V., Papadrakakis, M.: Metamodel-based computational techniques for solving structural optimization problems considering uncertainties. In: Tsompanakis, Y., Lagaros, N.D., Papadrakakis, M. (eds.) Structural Design Optimization considering Uncertainties. Structures and Infrastructures Series. Taylor & Francis, Taylor (2008)
16. Lee, D.S., Gonzalez, L.F., Periaux, J., Srinivas, K.: Robust design optimisation using multiobjective evolutionary algorithms. Comput. Fluids 37(5), 565–583 (2008)
17. Molina, J., Santana, L., Hernández-Díaz, A., Coello, C.A., Caballero, R.: g-dominance: Reference point based dominance for multiobjective metaheuristics. European Journal of Operational Research 197, 685–692 (2009)
18. Tsompanakis, Y., Lagaros, N.D., Papadrakakis, M.: Structural Design Optimization Considering Uncertainties. Structures & Infrastructures Series, vol. 1. Series Ed: Frangopol D.Taylor & Francis, USA (2008)
19. Whitley, D., Rana, S., Heckendorn, R.: Representation Issues in Neighborhood Search and Evolutionary Algorithms. In: Quagliarella, D., Périaux, J., Poloni, C., Winter, G. (eds.) Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, pp. 39–57. John Wiley & Sons, Chichester (1997)
20. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Eiben, A.E., et al. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
21. 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, CIMNE, pp. 95–100 (2002)

Multiobjective Dynamic Optimization of Vaccination Campaigns Using Convex Quadratic Approximation Local Search

André R. da Cruz[1], Rodrigo T.N. Cardoso[2], and Ricardo H.C. Takahashi[3]

[1] Mestrando em Engenharia Elétrica
Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil
andrercruz@cpdee.ufmg.br
[2] Departamento de Física e Matemática
Centro Federal de Educação Tecnológica de Minas Gerais,
Belo Horizonte, MG, Brasil
rodrigoc@des.cefetmg.br
[3] Departamento de Matemática
Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil
taka@mat.ufmg.br

Abstract. The planning of vaccination campaigns has the purpose of minimizing both the number of infected individuals in a time horizon and the cost to implement the control policy. This planning task is stated here as a multiobjective dynamic optimization problem of impulsive control design, in which the number of campaigns, the time interval between them and the number of vaccinated individuals in each campaign are the decision variables. The SIR (Susceptible-Infected-Recovered) differential equation model is employed for representing the epidemics. Due to the high dimension of the decision variable space, the usual evolutionary computation algorithms are not suitable for finding the efficient solutions. A hybrid optimization machinery composed by the canonical NSGA-II coupled with a local search procedure based on Convex Quadratic Approximation (CQA) models of the objective functions is used for performing the optimization task. The final results show that optimal vaccination campaigns with different trade-offs can be designed using the proposed scheme.

1 Introduction

The need to understand and to model the dynamics of disease proliferation has originated an area of research, the mathematical epidemiology [10]. It performs quantitative studies of the distribution of health/disease phenomena, besides assessing the effectiveness of interventions in public health [5]. This area offers important tools to analyze the spread and to execute control of infectious diseases. Epidemiology models can contribute to the design and analysis of epidemiological surveys, suggest crucial data that should be collected, identify trends, make general forecasts, and estimate the uncertainty in forecasts [13]. The model

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 404–417, 2011.

formulation process clarifies assumptions, variables, and parameters; moreover, models provide conceptual results such as thresholds, basic reproduction numbers, contact numbers, and replacement numbers. Mathematical models and computer simulations are useful tools for building and testing theories, assessing quantitative conjectures, answering specific questions, determining sensitivities to changes in parameter values, and estimating key parameters from data. Understanding the transmission characteristics of infectious diseases in communities, regions, and countries can lead to better approaches to the control of such diseases. Mathematical models are used in comparing, planning, implementing, evaluating, and optimizing various detection, prevention, therapy, and control programs [14,4].

The epidemiological model *SIR*, proposed by Kermack and McKendrick in 1927 [15], is one of the most used to represent infectious diseases. This model classifies the individuals in susceptible (S), infected (I) and recovered (R), and these states are related by a system of differential equations [1,14].

A big challenge in public health is related to the planning of vaccination campaigns, which aim to eradicate a disease, or to control its spread avoiding large epidemic peaks or significant endemic states, with the minimum possible cost. Many works have applied control theory to develop optimal vaccination strategies. Most of them used pulse vaccination, which is an impulsive control defined by the repeated application of a fixed vaccination ratio in discrete instants with equal time intervals [1,20,14,21,16,11,17,12,24].

In some previous works [6,7,8], the authors of this paper sketched some steps toward a multiobjective optimization approach for the design of the vaccination control of epidemics in a given time horizon. This approach is more flexible than former ones, since it allows different sizes of pulse control action in different instants, and it also allows the application of pulses at arbitrary time instants. However, in those works, the usage of conventional evolutionary computation algorithms as the optimization machinery conducted to unsatisfying results, probably due to the large dimension of the decision variable space, which is caused by the dynamic nature of the problem.

The present work proposes the use of a hybrid optimization machinery to cope with this problem: a canonical evolutionary multiobjective algorithm is coupled with a local search operator based on a Convex Quadratic Approximation (CQA) of the objective functions. The CQA is applied around some selected points of the nondominated population, in order to provide a local description of the objective functions that allows the accurate determination of good candidate points that are expected to be non-dominated solutions. The CQA employed here uses a semi-positive definite diagonal Hessian matrix [22].

The NSGA-II [9] is used as the basic multiobjective evolutionary algorithm, and the CQA operator to be embedded in the NSGA-II is based on the model presented in [18]. This CQA operator performs the minimization of the 1-norm of the error, considering a model that represents a lower bound for the function points, allowing the problem to be solved via linear programming. It must be pointed out that any other multiobjective evolutionary algorithm could be used. The CQA operator used here, however, is especially suitable for high dimensional functions. Other approximation schemes, such as the one presented in [22] would not work in the problem here, due to this high dimension.

To show the effectiveness of the proposed engine to the planning of vaccination campaigns, a Monte Carlo simulation experiment is developed, running the scheme with and without the CQA local search. It becomes clear the advantage of the hybrid method.

The paper is organized in the following way: Section 2 explains the SIR epidemics model; Section 3 shows the optimization model for the epidemics control problem; Section 4 presents the CQAs for function approximation; Section 5 shows the proposed optimization engine; Section 6 shows the results; and Section 7 concludes the paper.

2 The SIR Model

This work uses the deterministic SIR (Susceptible-Infected-Recovered) epidemics model to describe the dynamics of a disease in a population. This model classifies the individuals in three compartments related to their health condition:

- Susceptible (S): an individual which may be infected by the disease from the contact with infected individuals;
- Infected (I): an individual which has the disease and can transmit it to susceptible individuals through contact; and
- Recovered (R): an individual which was cured or received a vaccine, becoming immune to the disease.

The SIR model is represented by a system of three differential equations, shown in Equation (1).

$$\begin{aligned} \frac{dS}{dt} &= \mu N - \mu S - \frac{\beta I S}{N}, && S(0) = S_o \geq 0 \\ \frac{dI}{dt} &= \frac{\beta I S}{N} - \gamma I - \mu I, && I(0) = I_o \geq 0 \\ \frac{dR}{dt} &= \gamma I - \mu R, && R(0) = R_o \geq 0 \end{aligned} \tag{1}$$

In this system, S, I and R are respectively the number of susceptibles, infected and recovered, N is the total number of individuals (which is supposed to be fixed: $S(t)+I(t)+R(t) = N$, $\forall t$). The parameters are: β is the transmission rate, γ is the recovery rate of infected individuals, and μ is the rate of new susceptibles. To keep the number of individuals in population constant, the mortality rate and the birth rate are made equal. As N is constant along the time, the variables can be written as ratios: $s(t) = S(t)/N$, $i(t) = I(t)/N$, $r(t) = R(t)/N$, and $r(t) = 1 - s(t) - i(t)$.

This epidemics model examines the spread of disease in a population during a time-stage, in an average sense. It is mathematically and epidemiologically well-conditioned [14]. The basic reproductive number R_0 is defined as $R_0 = \beta/(\mu+\gamma)$, which represents the average number of infections produced by an infected individual. The article [14] shows that the system SIR has an asymptotically stable endemic equilibrium if and only if $R_0 > 1$.

The parameter values used for the purpose of buiding an example in this work are shown in Table 1. The number of individuals considered is $N = 1000$. The initial condition, in ratios, is $A = (s_o, i_o, r_o) = (0.99; 0.01; 0.00)$. The time horizon adopted is $T\_CNTRL = 360$ (u.t.).

Table 1. Parameters of SIR

Parameter	Value
μ	1/90
β	0.15
γ	1/50

Figure 1 shows the behavior of the system without intervention for this configuration. In this case, the stable point is $F = (0.21; 0.28; 0.51)$. Notice that the amount of infected individuals in the stable point (the endemic equilibrium) is very significant and, for this reason, it is necessary act to control the disease.

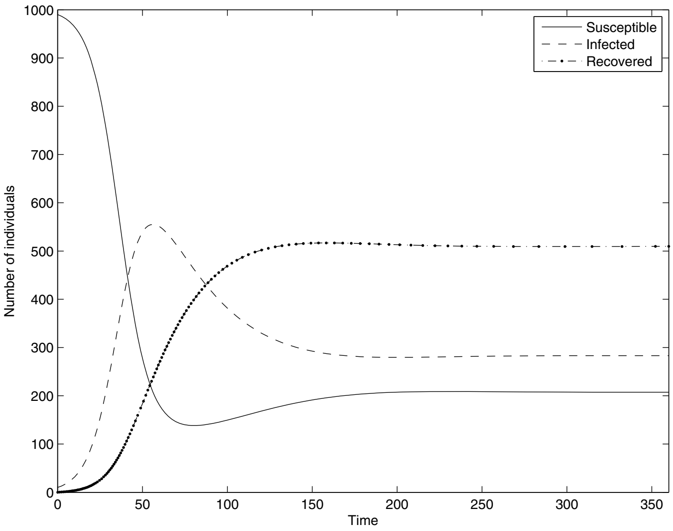


Fig. 1. Behavior of the SIR model without intervention, considering the parameters shown in Table 1

3 Optimization Model

This paper performs the planning of vaccination campaigns using a dynamic optimization approach, in which the SIR model is the dynamic system, and an arbitrary number of impulsive control inputs (vaccination campaigns), with arbitrary amplitudes, are applied in arbitrary time instants.

The main idea of impulsive control [23] is to split the continuous-time interval in some stages, performing impulsive control actions just in some time instants. The dynamic system keeps its autonomous dynamics in the time intervals between the consecutive impulsive control actions. The solution of the dynamic programming impulsive control problems is sought here in open-loop – using an evolutionary algorithm, which allows rather arbitrary objective function and constraints, and is an alternative to the use of enumerative algorithms, which have prohibitive computational complexity [3].

The time horizon $[0, T\_CNTRL]$ has to be partitioned in a set of time instants $\Gamma = \{\tau_0, \ldots, \tau_M\}$, such that $\tau_k < \tau_{k+1}$, $\tau_0 = 0$, $\tau_M = T\_CNTRL$, and it does not have to be equally spaced. The time instant τ_k^+ is defined as a time instant "just before" the impulsive action in τ_k. The following discrete-time variable notation is considered: $x[k] = x(\tau_k)$ and $x[k^+] = x(\tau_k^+)$, for each state $x = s, i$ or r.

The optimization variables of the problem are:

- The number of vaccination campaigns (stages): M,
- The vector of time instants between the campaigns: $\Gamma = \{\tau_0, \ldots, \tau_M\}$,
- The percentages of vaccines of the susceptible population during the campaigns: $P = \{p[1], \ldots, p[M]\}$, such that $p[k] = p(\tau_k)$, for each τ_k in Γ.

Notice that Γ depends on M, and P depends on Γ.

Since the vaccination acts only in the ratio of susceptible, then:

$$\begin{cases} s[k^+] = s[k](1 - p[k]); \\ i[k^+] = i[k]; \\ r[k^+] = r[k] + s[k]p[k]; \\ k = 0, 1, \ldots, M - 1. \end{cases} \tag{2}$$

Each initial value problem in (1) is valid during the time in which there is no control action (vaccination) in the system – the system presents its autonomous dynamics within such time intervals. A new initial condition for the system is established in each stage according to the difference equation (2), linking each stage with the next one. The set of initial conditions in $t = 0$ must be previously known, if the explicit simulation of the system is to be performed.

The constraints are modeled as:

- Each time interval between stages $\Delta\tau_k = \tau_{k+1} - \tau_k$ has to obey the relation: $0 < tmin \leq \Delta\tau_k \leq tmax < T\_CNTRL$;
- The sum of $\Delta\tau_k$ is not allowed to overcome the time horizon T_CNTRL; this determines the number of campaigns M;

- Each vaccination ratio $p[k]$ has to follow the rule: $0 < pmin \leq p[k] \leq pmax \leq 1.0$;
- It is desired that the number of susceptible and infected individuals, after $t = T\_INIC$ respectively, satisfy $S(t) \leq s\_tol, I(t) \leq i\_tol$, for $t \in (T\_INIC, T\_CNTRL)$.

The vaccination campaign planning is formulated as a multiobjective optimization problem with two objective functions that should be minimized:

- F_1 – the integral of infected population during the optimization horizon;
- F_2 – the total cost with the campaign (sum of vaccines added with a fanciful fixed cost of each campaign that represents, for instance, transportation, advertisement, etc.).

Therefore, the bi-objective optimization model can be formulated as in Problem (3):

$$\min_{M,\Gamma,P} \begin{cases} F_1 = \int_0^{T\_CNTRL} I(t)dt \\ \\ F_2 = c_1 \cdot M + c_2 \cdot \sum_{k=0}^{M} S[k] \cdot p[k] \end{cases} \tag{3}$$

subject to:

$$\begin{cases} \frac{dS}{dt} = \mu N - \mu S - \frac{\beta IS}{N}, \ S(0) = S_o \geq 0; \\ \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, \ I(0) = I_o \geq 0; \\ \\ \frac{dR}{dt} = \gamma I - \mu R, \ R(0) = R_o \geq 0; \\ \\ t \in (\tau_k^+, \tau_{k+1}]; \\ \\ s(\tau_k^+) = s[k^+] = s[k](1 - p[k]); \\ i(\tau_k^+) = i[k^+] = i[k]; \\ r(\tau_k^+) = r[k^+] = r[k] + s[k]p[k]; \\ k = 0, 1, \ldots, M-1; \\ \\ 0 < tmin \leq \Delta\tau_k \leq tmax < T\_CNTRL; \\ 0 < pmin \leq p[k] \leq pmax \leq 1.0; \\ \sum_{k=1}^{M} \Delta\tau_k \leq T\_CNTRL; \\ \\ S(t) \leq s\_tol, \ t \in (T\_INIC, T\_CNTRL); \\ I(t) \leq i\_tol, \ t \in (T\_INIC, T\_CNTRL); \\ \\ M \in \mathbb{N}^*. \end{cases}$$

The solutions of a multiobjective problem constitute the *non-dominated solution set*, or *Pareto-optimal set*. In a minimization problem with vector of objective functions $J \in \mathbb{R}^m$, if $y \in \mathbb{R}^n$ denotes the vector of decision variables of the problem and $D \subset \mathbb{R}^n$ the set of feasible solutions y, the Pareto-optimal set $Y \subset D$ is characterized by:

$$Y = \{\bar{y} \in D; \neg\exists\ y \in D : J_i(y) \leq J_i(\bar{y}),\quad \forall i = 1, \ldots, m; J(y) \neq J(\bar{y})\}. \tag{4}$$

The image of the Pareto-optimal set Y by the objective function J, or $J(Y)$, is the *Pareto-front*.

This paper deals with the problem of finding an estimate of the Pareto-optimal set of problem (3). The parameters considered here are: $T\_CNTRL = 360$ *u.t.*, $tmin = 5.0$ *u.t.*, $tmax = 20.0$ *u.t.*, $pmin = 0.4$, $pmax = 0.8$, $T\_INIC = 100$ *u.t.*, $s\_tol = 400$, $i\_tol = 5$, $c_1 = 1000$, $c_2 = 1$. The other constant values are in Table 1.

4 Convex Quadratic Approximation

Consider a real-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, with given distinct sample data $\{(\boldsymbol{x}^{(k)}, f(\boldsymbol{x}^{(k)}))\}$, for $k = 1, 2, \cdots, m$, where m is the number of available points to be used to find a convex quadratic real-valued function $q : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$q(\boldsymbol{x}) = h_0 + \nabla \boldsymbol{h}^T \boldsymbol{x} + 0.5 \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x}, \tag{5}$$

in which $\boldsymbol{x} \in \mathbb{R}^n$, $h_0 \in \mathbb{R}$, $\nabla \boldsymbol{h} \in \mathbb{R}^n$, $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ is at least semi-definite positive. The idea is that $f(\boldsymbol{x}^{(k)}) \cong q(\boldsymbol{x}^{(k)})$ as much as possible.

The convex quadratic approximation can be described as a function $q(\boldsymbol{p}, \boldsymbol{x})$, in which $\boldsymbol{p} \in \mathbb{R}^s$ is a vector of the approximation function parameters. It is assumed that enough data points are available so that $m \geq s$. In particular, it was adopted $m = 2s$ in this work. In the model with diagonal Hessian, the number of necessary parameters for the quadratic approximation is $s = 1 + n + n = 2n + 1$.

The convex quadratic approximation via linear programming uses the minimization of the error 1-norm to fit the data. The following properties hold:

- The 1-norm approximation is robust with respect to outliers [2,19].
- The 1-norm approximation always interpolates a subset of data points [2].
- The convex quadratic approximation represents a lower bound for the sample points, providing also an indication of the global trend of the original function.

The method generates a convex quadratic function of the form:

$$q(c_0, \nabla \boldsymbol{c}, \boldsymbol{D}; x) = c_0 + \nabla \boldsymbol{c}^T \boldsymbol{x} + 0.5 \boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x} = c_0 + \sum_{i=1}^{n} c_i x_i + 0.5 \sum_{i=1}^{n} d_i x_i^2, \tag{6}$$

$\boldsymbol{D}$ is a diagonal matrix with elements $d_1, \ldots, d_n$ in its principal diagonal, with $d_i \geq 0$ for all $1 \leq i \leq n$. This is a simple convex quadratic function. For a fixed

$\boldsymbol{x}$, q is linear in c_0, $\boldsymbol{c}$, and $\boldsymbol{D}$. Consequently, the general problem is reduced to a linear optimization problem [18]:

$$\min_{c_0,\boldsymbol{c},\boldsymbol{D}} \quad \sum_{k=1}^{m} \Big(f(\boldsymbol{x}^{(k)}) - q(c_0,\boldsymbol{c},\boldsymbol{D};\boldsymbol{x}^{(k)}) \Big) \tag{7}$$

subject to:

$$\begin{aligned} & f(\boldsymbol{x}^{(k)}) \geq q(c_0,\boldsymbol{c},\boldsymbol{D};\boldsymbol{x}^{(k)}), && \forall\, 1 \leq k \leq m \\ & d_i \geq 0, && \forall\, 1 \leq i \leq n \end{aligned}$$

The first constraint makes this approximation to become a lower bound for the sample points. This minimization problem has $2n+1$ variables and $m+n$ inequality constraints. This is a linear programming (LP) problem, which can be solved with efficient algorithms.

The approximated functions obtained in this way may be constructed within the multiobjective optimization procedure using samples produced around some individuals in the nondominated set, in some iterations, through the evolutionary process. As each approximation function is convex (even when the original is not), they can be aggregated into a weighted sum function, which is used for the purpose of making guesses about the location of the requested nondominated points. Preferentially, the objective function weights should be selected in order lead to a diverse front. This CQA leads to an improvement of the performance of evolutionary algorithms, with a slightly higher computational cost.

5 Optimization Engine

A canonical implementation of NSGA-II, denoted here by *CAN*, is used as the basis algorithm for performing the optimization. The hybrid algorithm, denoted here by *QAP*, is constructed taking the basis algorithm, with the following modification, that is executed each five generations:

- A set of r points is chosen from the current nondominated solution set;
- A set of m new points is generated randomly, with Gaussian distribution, around each point chosen in the last step;
- All objective functions are evaluated on the new points;
- A CQA is calculated for each objective function, for each one of the r sets of $m+1$ points;
- The quadratic approximation functions are used in order to generate p estimates of Pareto-optimal points, using the weighted-sum procedure with a quadratic programming algorithm (the p weights are considered as equally spaced between 0 and 1);
- The new $r \times p$ points are evaluated, and merged to the current population.

In the experiments conducted here it was used $r = 2$, with the deterministic choice of the two extremes solutions of the set. It is adopted $m = 2s$, in which

s is the number of parameters of the approximated convex quadratic function. The number of points generated for each set of approximated functions is chosen as $p = N/2$, in which N is the population size. This means that, each time the local search is conducted, N points are generated, and the selection procedure is performed over $3N$ points, $2N$ individuals coming from the usual iteration of NSGA-II and N coming from the local search.

The problem (3) was solved using both CAN and QAP. Each algorithm was tested in 30 independent executions. In both CAN and QAP versions, the population size was $popsize = 40$, the number of generations was $ng = 100$, the probability of crossover was $p_c = 0.8$, the probability of mutation was $p_m = 0.05$ and the distribution for crossover and mutation were $\eta_c = \eta_m = 10$.

For each generation, it was computed the S-metric in order to evaluate the convergence and quality of solutions returned by the evolutionary multiobjective algorithms. The S-metric values were normalized in each test by the highest value obtained. At the end of the tests, the solutions of all executions were merged and a new non-dominated sorting was performed over the pool of all solutions. The number of solutions remaining in the first Pareto front for each algorithm is counted.

6 Experiments and Results

The executions of the two versions of NSGA-II (with and without the CQA procedure) and the application of the methodology explained in Section 5 are shown in Figure 2.

The final nondominated set is shown in Figure 2, in which the axes represent the integral of infected individuals during the control, F_1, normalized by the total points returned by the Runge-Kutta method, and the cost of the control during the time horizon, F_2. The left side points correspond to more costly campaigns, with smaller numbers of infected people, while the campaigns situated at the right side represent less expensive alternatives. The blue cross points of Fig. 2 represent the 98 nondominated solutions returned by NSGA-II with local search (QAP) in all runs. The red asterisk points represent the 70 nondominated solutions returned by the canonical NSGA-II (CAN) in all runs.

To illustrate the actual vaccination policies that come from the proposed procedure, the final nondominated solution picked from the left side of the final nondominated front (Fig. 2) is shown in Fig. 3 and another one, picked from the right side of the final front, is shown in Fig. 4. These figures show the behavior of the controlled system, the percentage and total number of individuals vaccinated. Note that the vaccination pulses employed in the vaccination campaigns are allowed to vary from one campaign to the next one, and also that the intervals between campaigns are different too. This is possibly responsible for the gaps in the final nondominated front: each disconnected line corresponds to a set of policies concerning about a different number of campaigns.

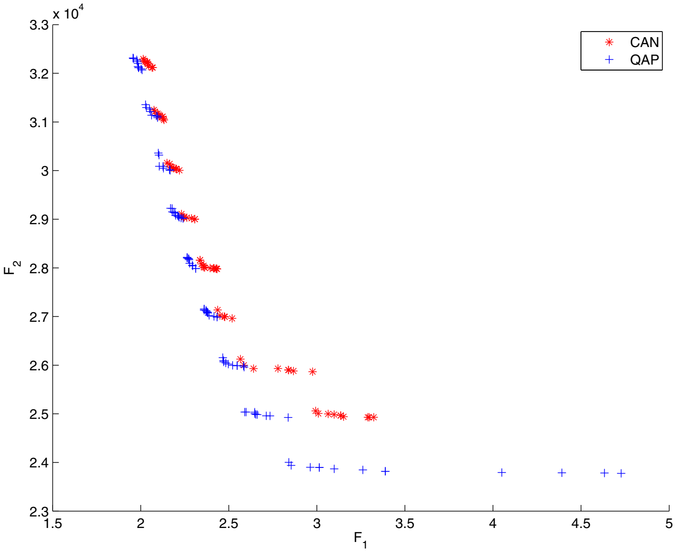


Fig. 2. Results of the Monte Carlo simulations explained in Section 5 to validate the proposed engine for cope to Problem 3. Final nondominated set for Problem 3. F_1 represents the integral of infected population during the control and F_2 represents the total cost of the vaccination control. The blue crosses represent the solutions returned by NSGA-II with local search (QAP) and the red asterisks represent the solutions returned by the canonical NSGA-II (CAN).

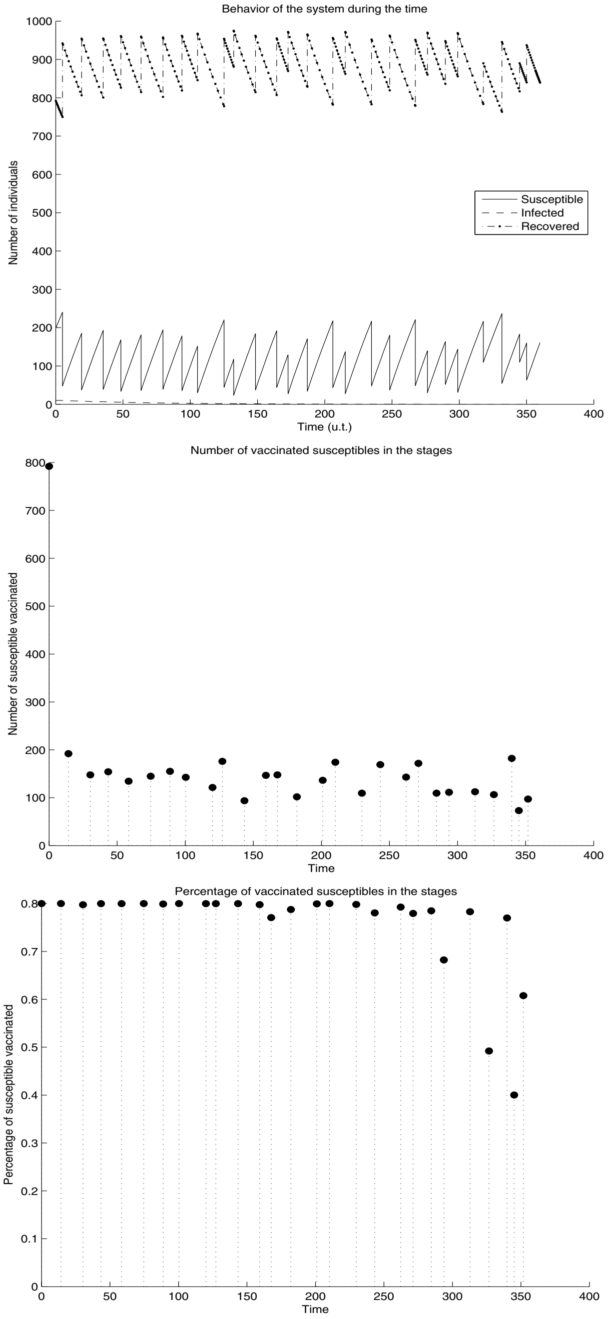


Fig. 3. Example of a Pareto-optimal solution picked from the left side of Figure 2. The first graphic shows the system variable time behavior, the second one shows the percentage of susceptible vaccinated during each campaign, and the third one shows the number of susceptibles vaccinated.

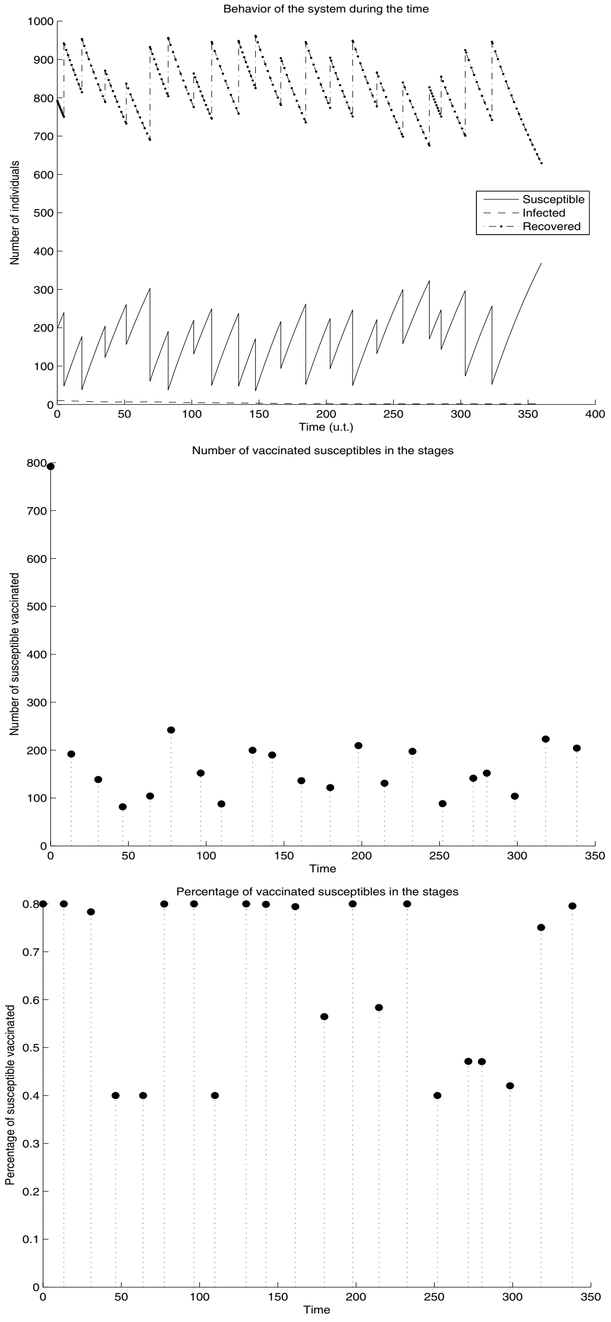


Fig. 4. Example of a Pareto-optimal solution picked from the right side of Figure 2. The first graphic shows the system variable time behavior, the second one shows the percentage of susceptible vaccinated during each campaign, and the third one shows the number of susceptibles vaccinated.

7 Conclusion

This work presented a multiobjective design methodology for vaccination policies which have the purpose of minimizing both the number of infected individuals in a population and the cost of vaccination campaigns. The problem is formulated as an impulsive open-loop dynamic optimization problem, in which the number of impulses, the amplitudes of such impulses, and the time instants in which they occur are the decision variables. The epidemics dynamic behavior is simulated using a SIR model.

The set of non-dominated policies was found using a hybrid NSGA-II, and also employs a CQA of the objective functions in order to provide a local search mechanism. The specific CQA employed here is based on the minimization of the 1-norm of the approximation error. The estimation of this CQA model is formulated as an LP problem. The resulting CQA model is suitable for the approximation of high dimensional functions – which allows the application of this technique in the specific problem treated here.

The main conclusion of this work is that a multiobjective optimization approach for determining "optimal" vaccination policies is interesting, since it provides a variety of different solutions that can be implemented in practice. The local search technique speeds up the convergence and the quality of the final nondominated policies, being fundamental for the correct determination of the Pareto-optimal solutions in this case.

Acknowledgments

The authors acknowledge the support from the Brazilian agencies Capes, CNPq and Fapemig.

References

1. Anderson, R.M., May, R.M.: Infectious diseases of humans: dynamics and control. Oxford University Press, Oxford (1992)
2. Barrodale, I.: L1 approximation and the analysis of data. Applied Statistics, 51–57 (1968)
3. Bertsekas, D.P., et al.: Dynamic programming and optimal control (1995)
4. Chubb, M.C., Jacobsen, K.H.: Mathematical modeling and the epidemiological research process. European Journal of Epidemiology 25(1), 13–19 (2010)
5. Clancy, D.: Optimal intervention for epidemic models with general infection and removal rate functions. Journal of Mathematical Biology 39(4), 309–331 (1999)
6. Cruz, A.R., Cardoso, R.T.N., Takahashi, R.H.C.: Uma abordagem multiobjetivo para o problema de controle biológico através da programação dinâmica não-linear via NSGA-II. In: Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional, pp. 44–55 (September 2007)
7. Cruz, A.R., Cardoso, R.T.N., Takahashi, R.H.C.: Uma abordagem multiobjetivo para o planejamento de campanhas de vacinação via NSGA-II. In: Anais do XL Simpósio Brasileiro de Pesquisa Operacional (September 2008)

8. Cruz, A.R., Cardoso, R.T.N., Takahashi, R.H.C.: Multi-objective design with a stochastic validation of vaccination campaigns. In: Preprints of the IFAC Workshop on Control Applications of Optimisation (CAO 2009), vol. 7(1), pp. 1–6 (2009)
9. 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 (August 2002), http://dx.doi.org/10.1109/4235.996017
10. Diekmann, O., Heesterbeek, J.A.P.: Mathematical epidemiology of infectious diseases: Model building, analysis, and interpretation. Wiley, Chichester (2000)
11. d'Onofrio, A.: Stability properties of pulse vaccination strategy in SEIR epidemic model. Mathematical Biosciences 179(1), 57–72 (2002)
12. d'Onofrio, A.: On pulse vaccination strategy in the SIR epidemic model with vertical transmission. Applied Mathematics Letters 18(7), 729–732 (2005)
13. Hethcote, H.W.: Three basic epidemiological models. Applied Mathematical Ecology, 119–144 (1989)
14. Hethcote, H.W.: The mathematics of infectious diseases. SIAM Review 42(4), 599–653 (2000)
15. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics-III. further studies of the problem of endemicity. Bulletin of Mathematical Biology 53(1), 89–118 (1991)
16. Lu, Z., Chi, X., Chen, L.: The effect of constant and pulse vaccination on SIR epidemic model with horizontal and vertical transmission. Mathematical and Computer Modelling 36(9-10), 1039–1057 (2002)
17. Moghadas, S.M.: Analysis of an epidemic model with bistable equilibria using the Poincaré index. Applied Mathematics and Computation 149(3), 689–702 (2004)
18. Rosen, J.B., Marcia, R.F.: Convex quadratic approximation. Computational Optimization and Applications 28(2), 173–184 (2004)
19. Rosen, J.B., Park, H., Glick, J., Zhang, L.: Accurate solution to overdetermined linear equations with errors using L1 norm minimization. Computational Optimization and Applications 17(2), 329–341 (2000)
20. Shulgin, B., Stone, L., Agur, Z.: Pulse vaccination strategy in the SIR epidemic model. Bulletin of Mathematical Biology 60(6), 1123–1148 (1998)
21. Stone, L., Shulgin, B., Agur, Z.: Theoretical examination of the pulse vaccination policy in the SIR epidemic model. Mathematical and Computer Modelling 31(4-5), 207–216 (2000)
22. Wanner, E.F., Guimarães, F.G., Takahashi, R.H.C., Fleming, P.J.: Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. Evolutionary Computation 16(2), 185–224 (2008)
23. Yang, T.: Impulsive control. IEEE Transactions on Automatic Control 44(5), 1081–1083 (2002)
24. Zaman, G., Kang, Y.H., Jung, I.H.: Stability analysis and optimal vaccination of an SIR epidemic model. BioSystems 93(3), 240–249 (2008)

Adaptive Technique to Solve Multi-objective Feeder Reconfiguration Problem in Real Time Context

Carlos Henrique N. de Resende Barbosa[1,2],
Walmir Matos Caminhas[2], and Joao Antonio de Vasconcelos[2]

[1] Universidade Federal de Ouro Preto,
Departamento de Ciencias Exatas e Aplicadas,
Rua 37, no 115, Loanda, ZIP 35931-026 - Joao Monlevade, MG, Brasil
[2] Universidade Federal de Minas Gerais,
Departamento de Engenharia Eletrica,
Avenida Antonio Carlos, 6627, Pampulha, ZIP 31270-010
Belo Horizonte, MG, Brasil
cbarbosa@decea.ufop.br, {caminhas,joao}@cpdee.ufmg.br

Abstract. This paper presents an innovative method to solve the reconfiguration problem in a distribution network. The main motivation of this work is to take advantage of the power flow analysis repetition when reconfiguration leads the network to a previous configuration due to cyclical loading pattern. The developed methodology combines an optimization technique with fuzzy theory to gain efficiency without losing robustness. In this methodology, the power flow is estimated by well-trained neo-fuzzy neuron network to achieve computing time reduction in the evaluation of individuals during evolutionary algorithm runs. It is noteworthy that the proposed methodology is scalable and its benefits increase as larger feeders are dealt. The effectiveness of the proposed method is demonstrated through examples. The overall performance achieved in the experiments has proved that it is also proper to real time context.

Keywords: multi-objective optimization, NSGA-II, fuzzy inference, feeder reconfiguration.

1 Introduction

Efficient reconfiguration of radial distribution power systems can save considerable amount of energy, since losses on their distribution lines are not negligible and equipments are faulty [1,2,3]. Energy losses can be reduced and system preparedness can be improved if distribution systems are reconfigured in accordance with seasonal loading. As Das [2] stated, potential savings straightly depends on network size. Through reconfiguration, gain opportunities increase with the number of maneuverable switches in the system [4]. In this work, reconfiguration means to change the on/off status of operational switches, which leads to distinct connection patterns among utility sources and customers.

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 418–432, 2011.

To perform optimization in feeder reconfiguration problem, we need to know all the steady state voltages at all buses. Hence, power flow analysis is needed and it is an important computing task to manage any distribution power system properly. In this way, an efficient method for power flow analysis has a critical role, since it takes time to find voltage values on each bus. Here, in particular, system optimization requires a fast and precise power flow analysis.

Feeder reconfiguration is a combinatorial problem with nondeterministic polynomial resolution time, since the search space has its size defined by N_S switches as being equal to 2^{N_S} possibilities [2,5]. Also, a multi-objective approach can model this same reconfiguration problem. Therefore, evolutionary algorithms are much appropriated to find optimal configuration solutions. In these algorithms, an individual evaluation requires high computational effort, since power flow analysis is done through iterative methods or equation system solvers. Due to network radiality, network size and loading, numerical analysis is better performed by iterative methods when conventional approach is implemented. The power flow computation method affects the overall algorithm runtime decisively. Real time application would demand a fast power flow resolution, mainly if we must deal with feeder restoration cases. In this paper, a proposal to alleviate processing and reduce overall runtime of the algorithm, capable of solving the feeder reconfiguration problem, is made. It simply consists on estimation of the network flow through an on-line fuzzy network, which corresponds to the individual evaluation step in the evolutionary multi-objective algorithm. Obviously, the fuzzy network must be previously trained through a set of samples, extracted from a conventional power flow method such as backward/forward sweep.

In this paper, we introduce a new methodology to identify optimal or near optimal configurations in radial distribution feeders, given some physical and technical constraints. The implemented algorithm works as a real-time tool, adopting combination between multi-objective optimization techniques and fuzzy logic approach in order to obtain good convergence and robust performance. The fitness function that guides the searching is composed by three objectives: loss minimization, voltage flatness, and minimum switching operation.

This work is organized as follows. In Section 2, related works are listed along with their main contributions. The feeder reconfiguration problem and the multi-objective approach with constraints are detailed by Section 3, where mathematical model is also presented. At Section 4, the fuzzy network architecture and its usefulness to the proposed methodology are explained. Implementation issues are discussed in Section 5. Finally, Section 6 describes computational experiments and concluding remarks are given in Section 7.

2 Related Works

In the literature, most of the first algorithms applied to the reconfiguration problem were heuristic-based techniques. Despite of their restriction about network type and specificities, those algorithms were very fast indeed [5,6]. Other algorithms used a blend of optimization and heuristics to efficiently determine switching maneuvers [7]. In recent years, evolutionary algorithms have been successfully applied to distribution system reconfiguration and they have found as many solutions as other previously employed techniques [8].

Fuzzy theory was already applied to reconfiguration problem on distinct ways. Das [2] has employed fuzzy logic as predictor function, and four objectives were modeled by means of fuzzy sets in a multi-objective approach. In that work, four objectives were associated with membership functions indicating the degree of satisfaction in each one. Here, fuzzy inference is applied to speed up evaluation of individuals and therefore it does not take part of the multiobjective expression. The multi-objective reconfiguration problem was already divided in two steps by Lee [9]: determination of non-dominated solution and fuzzy decision making. In that work, fuzzy reasoning was applied to multiple criteria (contingency preparedness, feeder residual margin, number of switching operations, amount of transferred living load considering relative weights among them) decision making in order to get the most appropriate restoration plan under a specific situation. Real power loss minimization and loading pattern improvement of the feeder was done by Sahoo [10]. A genetic algorithm was employed with mutation step, tuned by fuzzy reasoning, to reduce computing time and to avoid destruction of the radial property of any network configuration which was coded by a chromosome. A similar work, developed in [11] by Song, incorporated fuzzy reasoning into an evolutionary programming implementation in order to control the mutation process. Through heuristic information about the network, the fuzzy part of the algorithm adaptively adjusts parameters to generate valid candidates only. Also using fuzzy adaption coupled to evolutionary programming, Venkatesh [3] suggested an algorithm which found optimal reconfiguration solution to radial distribution systems obtained from fuzzified objectives. Power loss, voltage and current violations, and switching number were minimized through a fuzzy multi-objective formulation and the reconfiguration was solved by an enhanced genetic algorithm in [12]. A proper restoration plan was obtained in [13] from a fuzzy reasoning approach and the problem was treated as a composition of fuzzy objective functions and fuzzy constraints. In that work, fuzzy approach replaced imprecise heuristic rules normally performed by an experienced operator.

In this work, we propose a new application for fuzzy inference using it along with NSGA-II [14] to solve a multiobjective problem. The aim of this combination is to reduce computation time required to assess bus voltages on the changes in load demand. Here, fuzzy inference speed up the evaluation of a power system configuration.

3 The Multi-objective Problem

The problem has a combinatorial nature, as stated before, since it deals with on/off statuses of all maneuverable switches in the distribution system. In this work, the configuration with the flattest voltage profile (close to 1 *p.u.*), the lowest real power loss, and the maximum number of energized loads must be found in a multi-objective problem definition. Any network configuration can be described through a vector $\bar{x}$, which indicates all the N_S switch statuses ($\bar{x} = [x_1 x_2 .. x_{N_S}]$). Mathematically, the three mentioned objectives are described in what follows. We can describe total real power loss minimization as:

$$\min \left(\sum_{i=1}^{N_L} x_k * Re(Z_k) * |I_k|^2 \right) \quad , \quad \forall k \in N_L \quad , \tag{1}$$

where N_L is the total number of distribution lines or branches, $Re(Z_k)$ represents the real component of series impedance of the branch k, and I_k is the current flowing through this same branch. At the same time, minimum bus voltage deviation from nominal or normalized ($1\ p.u.$) value must be achieved in all N_B buses. In order to account this criterion, the index proposed by Sahoo [10], which requires simpler computations than the one found in [3], is used:

$$\min \left(\frac{1}{N_B} \sum_{k=1}^{N_B} |V_k - V_k^{nom}| \right) \quad , \tag{2}$$

where N_B is the total number of buses, V_k is the actual bus voltage at bus k and V_k^{nom} is the nominal value in this same bus. The third and last objective is the number of switching operations which must be kept minimal in order to extend their life expectancy [15] and avoid switching maneuver repetition:

$$\min \sum_{k=1}^{N_S} |x_k - x_{0k}| \quad , \tag{3}$$

where N_S corresponds to the total number of switches in the distribution system, $\bar{x}_0$ equals the initial or previous network configuration, x_k equals 1 (0) if the switch is on (off) and, finally, $|x_k - x_{0k}|$ equals the minimal distance between two given configurations, which means a quantity of switching operations needed to modify the current configuration into the new one. Additionally, restrictions must be addressed due to physical and technical limitations. The first restriction is related to network topology. Once we are dealing with a distribution system, the radial topology (tree) must be preserved

$$x_k^f \leq x_{k-1}^f \quad . \tag{4}$$

The switch status of a branch k (x_k^f) depends on the switch status of the immediate adjacent branch (x_{k-1}^f), nearer to the common source node f, where a substation exists. In this network type, the following condition always holds:

$$\bar{x} * N_{AL} = N_B - 1 \quad , \tag{5}$$

where N_{AL} represents the total number of active (living) distribution lines. All N_B buses must stay connected to assure that no island was created after reconfiguration. In this way, all loads must remain fed by one of the available substations during reconfiguration:

$$\sum_{f=1}^{N_A} y_k^f = 1, \forall k = 1, .., N_L \quad . \tag{6}$$

where y_k^f is a binary variable which indicates that a line k is fed by a substation f, and N_A corresponds to the number of substations. The last restriction ensures that no current exceeds nominal value at any line of the system:

$$x_k * |I_k| \leq I_k^{max}, \forall k \in N_L \quad . \tag{7}$$

Each branch k tolerates current flow I_k, lower than an upper limit denoted by I_k^{max}.

3.1 Component Modeling Issues

Unlike transmission systems, distribution systems typically have a radial topology, where power is provided by a single source on the top of the tree structure. The radial topology and its particular features, such as higher resistance/ reactance ratio in branches, cause ill-conditioning on such power systems and Newton-Raphson (or its variants) becomes unstable or inappropriate [13,16].

Most of the conventional power flow methods consider power demand as constant values based on peak load condition [17]. This should not be the case in real distribution systems, as power demand varies over a period, leading to a wrong or poor approximation of bus voltages. Consequently, this will probably imply in a non-optimal configuration provided by the evolutionary algorithm. As load demand frequently changes in a practical feeder, the actual configuration optimality is temporary. Then, loads can be better modeled as power demands with seasonal variation. The variation on reactive and real power demand is represented by a time series which depicts an equivalent 24-hours power consumption interval (see Figure 1). This interval could be any predefined period such as a week, a month or a year. In order to train a Neo-Fuzzy Neuron (NFN) network [18] (details are given in Section 4), each real power consumption series was randomly obtained with values lying between 50% and 400% times the nominal load ($[0.5 * P_k^{nom}, 4.0 * P_k^{nom}]$). Reactive power consumption time series was obtained in the same way, but values were limited to the range $[-2.0 * Q_k^{nom}, 2.0 * Q_k^{nom}]$. When actual feeders are simulated, measured data are more appropriated to train the fuzzy network used in this work.

Load flow algorithms apply known methods, such as fast decoupled, backward/forward sweep or current injection methods to determine power system state. Nevertheless, such methods demand high computational effort. The power flow method should be fast enough to enable the optimization process. In this work, we used the backward/forward sweep with current summation version as the conventional method due to its convergence behavior when radial networks are dealt. The current summation method was preferred over another iterative

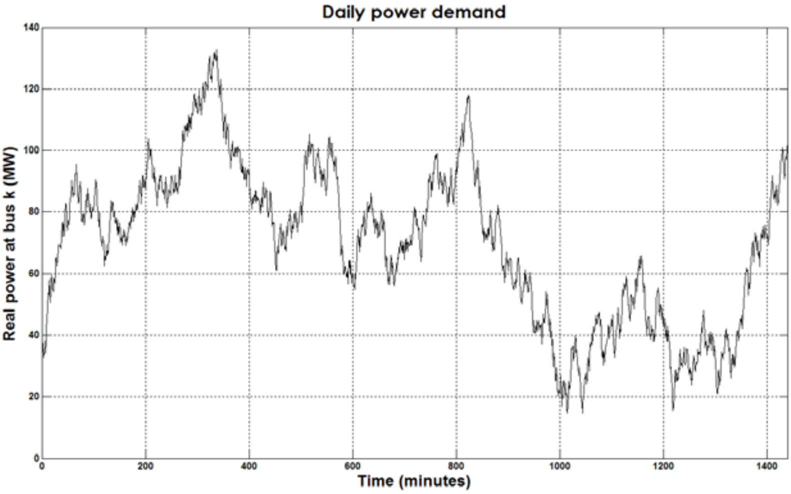


Fig. 1. An example of a random real power demand series

one, like power summation, due to its speed, relative robustness, and easy implementation. Moreover, it is faster than the power summation method, whereas the prior uses voltage V and current I instead of real power P and reactive power Q.

3.2 Multi-objective Discussion

The proposed methodology deal with the inherent combinatorial nature of the feeder reconfiguration problem and it is able to perform the optimization of the nonlinear multi-objective function. The switches statuses are the decision variable. Since there are only two statuses (on/off) for tie and sectionalizing switches, the solution space is discrete. Owing to this discretization, evolutionary techniques overcome classical techniques as their codification scheme straightly allows the treatment of ranges and set of valid values assumed by variables. The NSGA-II algorithm [14] with binary coded GA is used to find non-dominated solutions to the addressed multi-objective problem.

In this optimization problem, valid solutions are generally isolated one from each other. Unfeasible configurations are predominant in the discrete search space of the feeder reconfiguration problem. Therefore, crossover and mutation operators were carefully designed, since invalid solutions are numerous. As we chosen diferential codification to represent only the modified switches in relation to the current configuration, aiming reduction of the individual size, the crossover operator can be applied after chromossomes are rebuilt from the concise representation (diferential codification). Thus, genes belonging to two chromossomes are exchanged and validation is performed on the offspring to detect islanding and loops. Moreover, the mutation operator is carried out alternating the states of a pair of switches: tie and sectionalizing switches. Hence, a radial configuration preserves its topology. Still, elitism keeps one third of the best solutions.

To insert new candidate solutions in the current population, each of them is pre-analyzed, since preprocessing is justifiable due to the enormous search space created by the 2^{N_S} possibilities. Through this preprocessing occurring in selection, crossover and mutation steps, we assure that only valid configurations must be evaluated and evolved. The preprocessing basically consists of applying a simple heuristic to assure radiality on the configurations. Then, all starting guesses are valid. If each configuration evaluated by fitness function is feasible, then we ensure that only necessary computation has been done to find the optimal solution. After convergence, multiple solutions found by the evolutionary algorithm can be classified according to the operator's experience and preferences. For example, switching operations far from the root node can be prioritized in the decision process in order to avoid major disturbances caused by transfer of large amount of loads between large areas of the distribution system. On the other hand, reconfiguration must be done as soon as possible in restoration cases, with minimum number of switching operations, giving high priority to the automated switches [19].

4 Fuzzy State Estimation

In reconfiguration problem, solution evaluation can tolerate an approximation error generated by a power flow method. So, an accurate solution obtained from a rigid approach power flow method can bring no noticeable gain in terms of optimal configuration discovering.

Equivalent bus voltage estimation was implemented in [20] using artificial neural network. In this work, simple but efficient approximator fuzzy inference system architecture is adopted: the Neo-fuzzy neuron (NFN) network [18]. This architecture behaves as a universal approximator with the capability of detection and reproduction of nonlinearities between output and input variables. Every set of rules can be distinctly assigned to each input v_i. Each rule can contain only one premise, whose firing strength $\mu_{ij}(v_i)$ is equal to the membership degree of the input v_i associated to the output u_i. The rule set size does not depend on the number of inputs or membership functions per input. Thus, inputs are decoupled from each other in this architecture and each input has its own rules which can reduce the total rule quantity. Elementary membership function, like triangular or trapezoidal, is normally used, because they simplify calculation of the equivalent output. In a NFN network, a rule R has the following syntax:

$$\begin{array}{ccccccccc} R_i^1 & \longrightarrow & \text{If} & v_i & \text{is} & A_{i1}, & \text{then} & u_i & \text{is} \quad w_{i1} \\ & & & & & \vdots & & & \\ R_i^m & \longrightarrow & \text{If} & v_i & \text{is} & A_{im}, & \text{then} & u_i & \text{is} \quad w_{im} \end{array}$$

The consequent part of a rule is composed by an independent term (w_{ij}) similar to the zero-order TSK model [21]. This term is a weight which needs to be adjusted and it indicates how strong is the influence of some rule to a given network output. The NFN network is built from parallelized and redundant structures and thus it is fault-tolerant. Our implementation takes advantage of this aspect to speed up processing inside a multiprocessor environment.

In the implemented methodology, Neo-fuzzy Neuron networks gradually replace conventional power flow techniques in order to reduce computational burden without, however, deteriorate the solution obtained by conventional power flow analysis. On distribution system, real and reactive power demands generally behave in a seasonal manner, so fuzzy networks is successfully employed to simplify the bus voltage estimation process, achieving low deviation from the actual values. The NFN parameters stored in database are a, b, c, v_{min}, *delta* and q. A group of parameterized membership functions fill the interval corresponding to the universe of discourse V. Each triple (a,b,c) defines a triangular membership function (see Figure 2). Using those overlapped unit height functions in a complementary manner, shown in Figure 2, few rules are activated (corresponding to nonzero membership values) and easily detected. As a consequence, output computation becomes extremely fast [20]. The parameter v_{min} is the lower bound of the universe of discourse V, and delta equals the distance between two consecutives functions. Finally, the parameter q_i is a constant term of the rule m_{ij} associated with the output u_i.

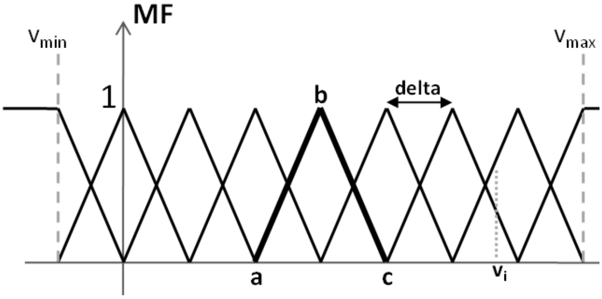


Fig. 2. Overlapped triangular membership functions over the interval X and NFN parameters significance

The input vector $\bar{v}$ consists of $2{*}NB{+}2$ variables: real power (P_C^k) and reactive power (Q_C^k) for each bus, total real power (P_T), and total reactive power (Q_T). The output variables are the voltage magnitude ($|V_k|$) and the voltage angle (θ_k) for each bus of the feeder, represented as:

$$\bar{u} = [V_1 \quad \theta_1 \ \ldots \ V_{NB} \quad \theta_{NB}]$$

As input vector keeps its size with the same variables, it is not affected by the network size, and fuzzy network overall performance almost remains constant.

5 The Implemented Algorithm

The goal is the implementation of an adaptive evaluation mechanism combining fuzzy inference with conventional calculus. During fitness evaluation, power flow analysis is speeded up, since there is a progressive exchange of the whole iterative processing for configuration fuzzy parameters storage in a database. Every time the fitness function is about to be called, the database is checked. If a previous record related to the network configuration under analysis exists, we must obtain the voltage values (magnitude and angle) from the NFN output, adjusted to each bus of the analyzed configuration. Otherwise, voltage values must be found using backward/forward sweep or another conventional power flow method. Still, if there are enough pair samples (power demand, bus voltage), a NFN network can be trained in parallel. Then, a new record containing the fuzzy network coefficients is added to the database.

To estimate voltage magnitude and angle at N_B buses of the feeder, $2 * N_B$ NFN networks must be employed. Indeed, we made a compensation for memory space requirement by using simpler repetitive calculation. Despite of required memory space be proportional to N_B buses and 2^{N_S} configuration possibilities to store NFN parameters, processing time spent to compute NFN outputs does not considerably change from one configuration to another. Better said, NFN output computing time does not alter, because its structure keeps the same. Figure 2 illustrates how configurations are evaluated inside the modified fitness function procedure. Considering the seasonal behavior of power demand in a real distribution feeder, power samples can be collected in each bus of the feeder along with their correspondent complex voltage, until sample size reaches a minimum default threshold value. At this moment, the NFN training can be triggered and the parameters found to each bus in the current configuration are stored as a new record in a database. In this database, a record consists of all the NFN parameters (see details in Section 4). Each record must be exclusively identified by a binary sequence formed by the joint of the bits which indicate the switches status. Listing 1 contains the main steps to evaluate a configuration. Figure 3 shows schematically how database is implemented.

Listing 1. Individual fitness evaluation steps.

```
procedure Evaluate Configuration Set
   1. Take one valid configuration from the repository.
   2. If there is a record in database, then load NFN parameters.
      Go to step 4.
   3. Perform power flow analysis. If sample size have reached
      a trigger value, execute NFN training. Next, insert the NFN
      parameters on database.
   4. Evaluate fitness function for the chosen configuration.
   5. Is there any remainder individual not yet evaluated? If
      so, return to step 1.
   6. Return fitness values.
```

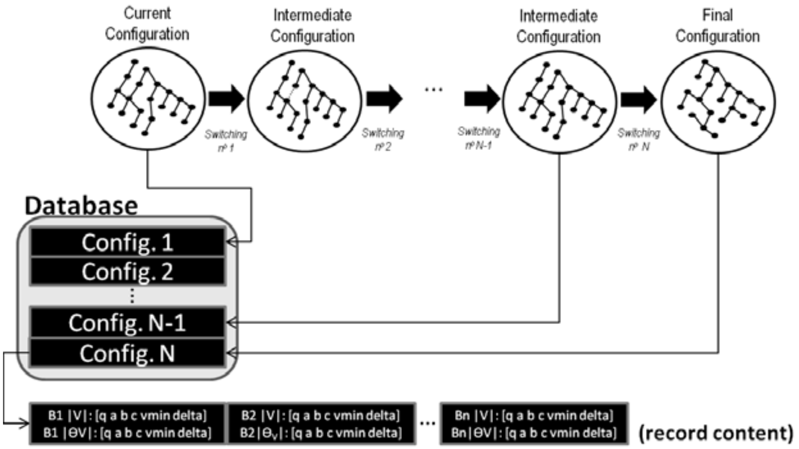

Current Configuration
Intermediate Configuration
Intermediate Configuration
Final Configuration
Switching nº 1
Switching nº 2
...
Switching nº N-1
Switching nº N
Database
Config. 1
Config. 2
Config. N-1
Config. N
B1 |V| : [q a b c vmin delta]
B1 |ΘV| : [q a b c vmin delta]
B2 |V| : [q a b c vmin delta]
B2 |Θ_V| : [q a b c vmin delta]
...
Bn |V| : [q a b c vmin delta]
Bn |ΘV| : [q a b c vmin delta]
(record content)

Fig. 3. A scheme of the database used to maintain NFN parameters

Sample size can be proportional to feeder size. Whenever a minimum number of samples has been attained, an update of the associated record in the database may occur. After that, the fitness function calls the fuzzy network instead of the backward/forward sweep method or another conventional method. In a given moment, reconfiguration may be worthless in achieving considerable improvements to the real power loss and voltage flat profile. In these cases, feeder must maintain its configuration. Still, updates can occur in the correspondent database record, which stores the NFN parameters of the actual configuration, to improve future fuzzy estimations.

Transition between some current configuration and the next, obtained from the non-dominated solution set, may be different depending on network loading. Keeping a database of NFN parameters becomes a major advantage, since a past power flow analysis could be needed again, when network went back to a given configuration. Therefore, it is expected that the benefits from the proposed methodology become more evident as more trained configurations are added to the database. To improve this methodology, an adaptive procedure could be incorporated to remove records stored in database which correspond to useless configuration analyzed in a recent past during a whole load demand cycle. The most preferable configurations should be identified and conveniently kept due to its recent fitness grade.

6 Experimental Results

Experiments were performed with three feeders already analyzed in the literature. It is assumed for all of them that every branch has a sectionalizing or a tie switch. The 11 kV 34-bus feeder tested was first presented by Chis [1]. The original 34-bus feeder has 33 distribution lines, four laterals, and one single source, all set as a radial network. Its initial power loss and voltage deviation index are 222.3 kW and 0.052 *p.u.*, respectively. Here, it was modified to admit reconfiguration as it can be seen in Figure 4. The initial configuration of this feeder is identical to the original feeder analyzed by Chis. And dashed lines represent inactive lines (their switches are off). A simulation demonstrates the effectiveness of the suggested methodology in terms of computing time and solution. Up to 400% load variation around nominal values in PQ buses were tried in a 34-bus test feeder, as stated in Section 3.1, in order to train the NFN networks. First, Table 1 contains complex bus voltages yielded by the backward/forward sweep (BFS) and fuzzy inference (FI) for the initial configuration. The Baran-Wu Method (BWM) [22] is taken to be the reference of both methods. The BFS method used a maximum error tolerance of 1×10^{-6} as a stop criterion. Comparing values found by both methods, a maximum discrepancy lower than 0.1% was achieved by the fuzzy inference system. As we trained the NFN network with increasing epoch number, the average error of the inference approximation diminished, but it caused no noticeable changes in the final non-dominated solution set. Discrepancies in magnitude voltages values found by BFS and FI methods are shown in Table 2. Although the average BFS error is 10 to 50 times lower than FI error, it can be observed a spread tendency on error values calculated for the BFS method, from the source node towards leaves (see Figure 5). In NFN approximation, magnitude estimation errors are not affected by adjacency among buses, which is verified by values found in Table 2. There is an intrinsic error related to the universal approximation provided by the fuzzy network and its training process, but it expresses as a uniform deviation.

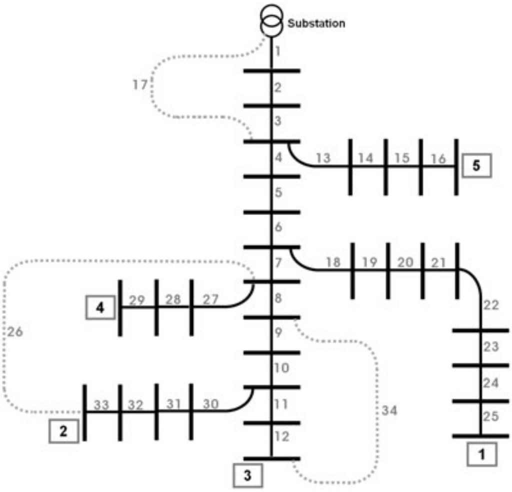


Fig. 4. The 34-bus test feeder modified to reconfiguration context

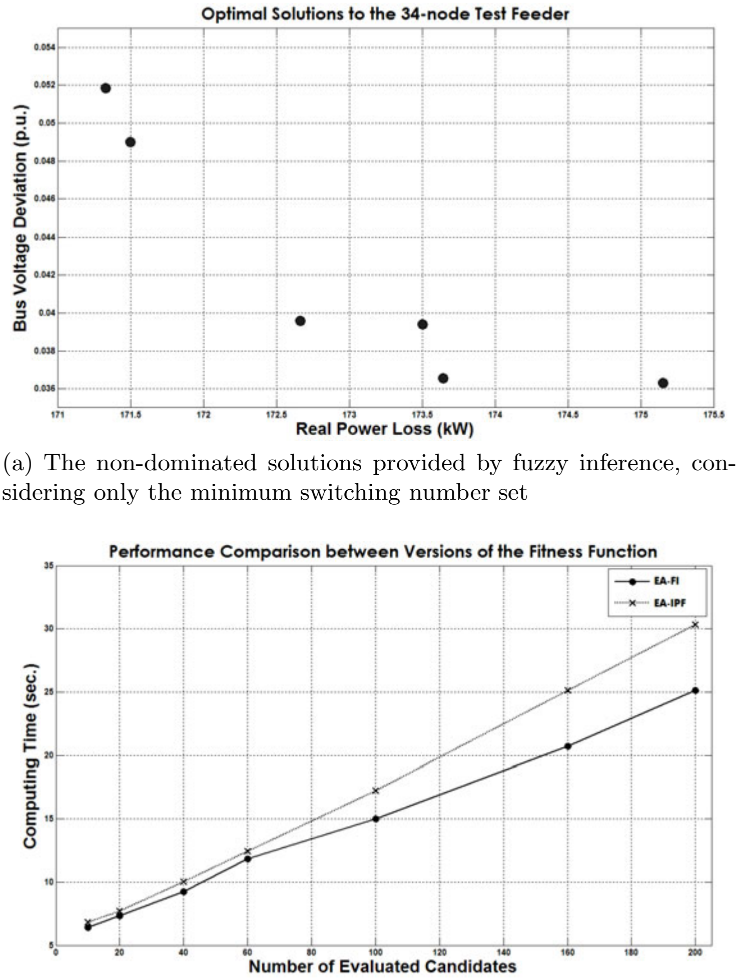
Optimization results using evolutionary algorithm with fuzzy inference (EA-FI) is presented by Figure 5(a). Taking into account the six optimal solutions found, an average total real power loss of 172.2 kW is computed (meaning a reduction of 9.9% in relation to the initial configuration).

Table 1. Complex bus voltages (magnitude and angle) found by backward/forward sweep (BFS) and fuzzy inference (FI) for the initial configuration of the 34-bus test feeder

B	BWM		BFS		FI		B	BWM		BFS		FI	
U	$\|V_k\|$	θ_k	$\|V_k\|$	θ_k	$\|V_k\|$	θ_k	U	$\|V_k\|$	θ_k	$\|V_k\|$	θ_k	$\|V_k\|$	θ_k
S	(p.u.)	(deg.)	(p.u.)	(deg.)	(p.u.)	(deg.)	S	(p.u.)	(deg.)	(p.u.)	(deg.)	(p.u.)	(deg.)
1	1.0000	0.00	1.0000	0.00	1.0003	0.00	18	0.9622	0.55	0.9622	0.55	0.9622	0.55
2	0.9941	0.05	0.9941	0.05	0.9942	0.05	19	0.9581	0.63	0.9581	0.63	0.9581	0.63
3	0.9890	0.10	0.9890	0.10	0.9892	0.10	20	0.9548	0.70	0.9548	0.70	0.9548	0.70
4	0.9820	0.22	0.9820	0.22	0.9821	0.22	21	0.9520	0.76	0.9520	0.76	0.9522	0.76
5	0.9760	0.32	0.9760	0.32	0.9762	0.32	22	0.9487	0.84	0.9487	0.84	0.9488	0.84
6	0.9704	0.41	0.9704	0.41	0.9706	0.41	23	0.9460	0.90	0.9460	0.90	0.9462	0.90
7	0.9666	0.50	0.9666	0.50	0.9667	0.50	24	0.9435	0.96	0.9435	0.96	0.9436	0.96
8	0.9644	0.56	0.9644	0.56	0.9648	0.56	25	0.9423	0.99	0.9423	0.99	0.9423	0.99
9	0.9620	0.62	0.9620	0.62	0.9620	0.62	26	0.9418	1.01	0.9418	1.01	0.9421	1.01
10	0.9608	0.65	0.9608	0.65	0.9613	0.65	27	0.9417	1.01	0.9417	1.01	0.9420	1.01
11	0.9603	0.66	0.9603	0.66	0.9604	0.66	28	0.9662	0.51	0.9662	0.51	0.9664	0.51
12	0.9602	0.66	0.9602	0.66	0.9605	0.66	29	0.9660	0.52	0.9660	0.52	0.9662	0.52
13	0.9887	0.11	0.9887	0.11	0.9888	0.11	30	0.9659	0.52	0.9659	0.52	0.9661	0.52
14	0.9884	0.12	0.9884	0.12	0.9889	0.12	31	0.9604	0.65	0.9604	0.65	0.9608	0.65
15	0.9883	0.12	0.9883	0.12	0.9884	0.12	32	0.9601	0.66	0.9601	0.66	0.9604	0.66
16	0.9883	0.12	0.9883	0.12	0.9884	0.12	33	0.9599	0.67	0.9599	0.67	0.9608	0.67
17	0.9659	0.49	0.9659	0.49	0.9660	0.49	34	0.9599	0.67	0.9599	0.67	0.9599	0.67

Table 2. Magnitude voltage differences found by comparing BFS and FI methods with reference values (BWM)

	BFS	FI		BFS	FI
Bus	$\|\delta V_k\|$	$\|\delta V_k\|$	Bus	$\|\delta V_k\|$	$\|\delta V_k\|$
	(%)	(%)		(%)	(%)
1	0.0000	0.0321	18	0.0011	0.0072
2	0.0001	0.0075	19	0.0011	0.0128
3	0.0002	0.0184	20	0.0010	0.0070
4	0.0004	0.0049	21	0.0009	0.0216
5	0.0005	0.0118	22	0.0010	0.0097
6	0.0008	0.0258	23	0.0012	0.0174
7	0.0011	0.0140	24	0.0011	0.0125
8	0.0009	0.0358	25	0.0010	0.0057
9	0.0009	0.0077	26	0.0013	0.0348
10	0.0010	0.0526	27	0.0014	0.0377
11	0.0010	0.0111	28	0.0010	0.0188
12	0.0009	0.0323	29	0.0012	0.0194
13	0.0002	0.0105	30	0.0012	0.0275
14	0.0004	0.0524	31	0.0009	0.0367
15	0.0007	0.0145	32	0.0010	0.0365
16	0.0007	0.0103	33	0.0010	0.0875
17	0.0009	0.0070	34	0.0011	0.0051



(a) The non-dominated solutions provided by fuzzy inference, considering only the minimum switching number set

(b) Evaluation time for the evolutionary algorithm with iterative power flow (EA-IPF) and fuzzy inference (EA-FI).

Fig. 5. The 34-bus test feeder

From Figure 5(b), we note an increasing difference in computing time between evolutionary algorithm with iterative power flow (EA-IPF) and fuzzy inference (EA-FI). Expressive results were obtained when larger populations were used in the evolutionary algorithm in terms of runtime.

The second system is a 33-bus feeder. Load data, transmission line details and tie lines available for switching are presented in [3]. After optimization, we found one optimal configuration with lines 7, 9, 14, 32, and 37 switched out. The minimum total real power loss achieved was 139.83 kW (33.8% reduction). Table 3 contains a comparison between backward/forward sweep and fuzzy inference system concern to computing time and error estimation provided by NFN. The third system is a 70-bus feeder, whose details and loading characteristics are

Table 3. Average computing time (ACT)/configuration and voltage magnitude approximation error of the fuzzy inference (FI) for the analysed distribution systems. Reference values are those obtained with backward/forward sweep (BFS).

Feeder	NFN		BFS
System	ACT (sec.)	Error (%)	ACT (sec.)
34 bus [1]	0.039	0.011	0.049
33 bus [3]	0.076	0.018	0.089
74 bus [12]	0.114	0.027	0.183

given in [12]. For small feeders, the proposed method shows minor improvement in comparison to EA-IPF. However, experiments have indicated that methodology can be promising when dealing with larger systems. It is important to note that fuzzy inference approximation is not itself the main concern, but maintenance of relative positions among configurations in terms of their fitness grades.

7 Conclusion

In this paper, the practical problem of feeder reconfiguration was dealt. The implemented algorithm demonstrated evident improvements in computing time, when individual fitness evaluation was totally replaced by fuzzy estimation during the process of multi-objective optimization. Identical optimal configuration solution was reached by both strategies, conventional and implemented evaluation, in the tests performed, showing a mean speed up ratio ranging from 1.1 to 1.5. Fuzzy estimation outperforms conventional power flow analysis when they are combined with the evolutionary algorithm. Being straightly extended to larger systems, the proposed methodology can be applicable to real time context, since NFN outputs can be instantaneously processed.

Acknowledgments. This work has been supported in part by CAPES and CNPq.

References

1. Chis, M., Salama, M.M.A., Jayaram, S.: Capacitor Placement in Distribution Systems Using Heuristic Search Strategies. IEE Proceedings on Generation, Transmission and Distribution 144(3), 225–230 (1997)
2. Das, D.: Reconfiguration of Distribution System using Fuzzy Multi-objective Approach. International Journal of Electrical Power & Energy Systems 28(5), 331–338 (2006)
3. Venkatesh, B., Ranjan, R.: Optimal Radial Distribution System Reconfiguration using Fuzzy Adaptation of Evolutionary Programming. Electrical Power & Energy Systems 25, 775–780 (2003)
4. Kalesar, B.M., Seifi, A.R.: Fuzzy Load Flow in Balanced and Unbalanced Radial Distribution Systems incorporating Composite Load Model. Electrical Power & Energy Systems 32(1), 17–23 (2010)
5. Zhou, Z.Q., Shirmohammadi, D., Liu, W.-H.E.: Distribution Feeder Reconfiguration for Service Restoration and Load Balancing. IEEE Transactions on Power Systems 12(2), 724–729 (1997)

6. Aoki, K., Nara, K., Itoh, M., Satoh, T., Kuwabara, H.: A New Algorithm for Service Restoration in Distribution Systems. IEEE Transactions on Power Delivery 4(3), 1832–1839 (1989)
7. Nara, K., Mishima, Y., Satoh, T.: Network Reconfiguration for Loss Minimization and Load Balancing. In: Power Engineering Society General Meeting, Ibaraki Univ., Japan, pp. 2413–2418. IEEE, Los Alamitos (2003)
8. Sarfi, R.J., Salama, M.A., Chikhani, A.Y.: A Survey of the State of the Art in Distribution System Reconfiguration for System Loss Reduction. Electric Power Systems Research 31(1), 61–70 (1994)
9. Lee, S.J., Lim, S.I., Bokk-Shin, A.: Service Restoration of Primary Distribution Systems based on Fuzzy Evaluation of Multi-criteria. IEEE Transactions on Power Systems 13(3), 1156–1162 (1998)
10. Sahoo, N.C., Ranjan, R., Prasad, K., Chaturvedi, A.: A Fuzzy-tuned Genetic Algorithm for Optimal Reconfigurations of Radial Distribution Network. Wiley InterScience, European Transactions on Electrical Power 17(1), 97–111 (2007)
11. Song, Y.H., Wang, G.S., Johns, A.T., Wang, P.Y.: Distribution Network Reconfiguration for Loss Reduction using Fuzzy controlled Evolutionary Programming. In: IEE Proceedings on Generation, Transmission and Distribution, vol. 144(4), pp. 345–350 (July 1997)
12. Huang, Y.C.: Enhanced Genetic Algorithm-based Fuzzy Multi-objective Approach to Distribution Network Reconfiguration. In: IEE Proceedings on Generation, Transmission and Distribution, vol. 149(5), pp. 615–620 (September 2002)
13. Hsu, Y.Y., Kuo, H.C.: A Heuristic based Fuzzy Reasoning Approach for Distribution System Service Restoration. IEEE Transactions on Power Delivery 9(2), 948–953 (1994)
14. 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)
15. Naga Raj, B., Prakasa Rao, K.S.: A new Fuzzy Reasoning Approach for Load Balancing in Distribution System. IEEE Transactions on Power Systems 10(3), 1426–1432 (1995)
16. Hsiao, Y.T., Chien, C.Y.: Enhancement of Restoration Service in Distribution Systems Using a Combination Fuzzy-GA Method. IEEE Transactions on Power Systems 15(4), 1394–1400 (2000)
17. Lopez, E., Opazo, H., Garcia, L., Bastard, P.: Online Reconfiguration considering Variability Demand: Applications to Real Networks. IEEE Transactions on Power Systems 19(1), 549–553 (2004)
18. Yamakawa, T., Uchino, E., Miki, T., Kusabagi, H.: A Neo Fuzzy Neuron and its Applications to System Identification and Predictions to System Behavior. In: Proceedings of the 2nd IIZUKA, IIizuka, Japan, pp. 477–483 (1992)
19. Pereira, M.A., Murari, C.F., Castro Jr., C.A.: A Fuzzy Heuristic Algorithm for Distribution Systems' Service Restoration. Fuzzy Sets and Systems 102(1), 125–133 (1999)
20. Hsu, Y.Y., Yang, C.C.: Fast Voltage Estimation using Artificial Neural Network. Electrical Power System 27(11), 1–9 (1993)
21. Jang, J.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Inteligence. Prentice Hall, New Jersey (1997)
22. Baran, M., Wu, F.: Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing. IEEE Transactions on Power Delivery 4(2), 1401–1407 (1989)

Variable Neighborhood Multiobjective Genetic Algorithm for the Optimization of Routes on IP Networks

Renata E. Onety[1], Gladston J.P. Moreira[2],
Oriane M. Neto[1], and Ricardo H.C. Takahashi[3]

[1] Universidade Federal de Minas Gerais, School of Electrical Engineering,
Belo Horizonte, MG 31270-901, Brazil

[2] Universidade Federal dos Vales do Jequitinhonha e Mucuri, Campus Mucuri,
Teófilo Otoni, MG 39803-371, Brazil

[3] Universidade Federal de Minas Gerais, Dep. of Mathematics, Belo Horizonte,
MG 31270-901, Brazil

Abstract. This paper proposes an algorithm to optimize multiple indices of Quality of Service of Multi Protocol Label Switching (MPLS) IP networks. The proposed algorithm, the Variable Neighborhood Multiobjective Genetic Algorithm (VN-MGA), is a Genetic Algorithm based on the NSGA-II, with the particular feature that different parts of a solution are encoded differently, at Level 1 and Level 2. In order to improve the results, both representations are needed. At Level 1, the first part of the solution is encoded, by considering as decision variables, the arrows that form the routes to be followed by each request (whilst the second part of the solution is kept constant), whereas at Level 2, the second part of the solution is encoded, by considering as decision variables, the sequence of requests, and first part is kept constant. The preliminary results shown here indicate that the proposed approach is promising, since the Pareto-fronts obtained by VN-MGA dominate the fronts obtained by fixed-neighborhood encoding schemes. Besides the potential benefits of the application of the proposed approach to the optimization of packet routing in MPLS networks, this work raises the theoretical issue of the systematic application of variable encodings, which allow variable neighborhood searches, as generic operators inside general evolutionary computation algorithms.

Keywords: Routing on IP Networks, Variable Neighborhood Search, Multi-objective Genetic Algorithm.

1 Introduction

With the emergence of new technologies, the transmission of multimedia applications has become an achievable goal. The new applications such as videoconferences, Video on Demand (VoD) or Voice over IP (VoIP) brought the need of some guarantees of network characteristics with respect to the quality of the

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 433–447, 2011.

data flow, such as minimum bandwidth or maximum delay [7]. However, in the conventional internet traffic, it is not possible to predict the path of the packets of transmission, i.e, there is no guarantee of the regularity of communication. For this reason, there are some mechanisms for Quality of Service (QoS) that allow differentiation of the flows transmitted and the definition of conditions in order to reach a level of quality from the prioritization of different flows according to their characteristics and objectives [18].

Recently, several technologies have been proposed in order to identify the type of information on IP networks, allowing the QoS requirements. The MPLS (Multi Protocol Label Switching) is an example of an alternative that makes possible the explicit routing of packets, which facilitates the provisioning of QoS according to the requirements of multimedia applications. This technology allows the addition of labels to packets in order to identify them.

Several studies have been proposed recently in order to develop an approach of Traffic Engineering for Routing with QoS. According to RFC-3272 (Request for Comments 3272), the Internet Traffic Engineering is defined as that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks [3]. Many of these studies deal with routing on IP networks and MPLS, using single-objective GAs [15,2] or deterministic methods, like Lagrangian Relaxation [9]. As the model of these studies is formulated with a single objective, the search can be biased to a certain goal, leading to solutions that are unsuitable under other objective viewpoint. For this reason, the multi-objetive strategies have received some attention. However, the use of multi-objective methods applied to the problem of routing on IP networks is not so extensive. Even so, most of the works execute the optimization of two parameters [1] and some studies use deterministic methods [11]. The current study deals with the optimization of three parameters which, according to [22], render the problem NP-complete. Thus, the techniques based on non-deterministic heuristics are likely to be the most suitable ones. Santos [20] proposes a dynamic evaluation for routing in an ambient of MPLS using multi-objective techniques. That research represents an initial reference for the present work, presenting the same objective functions.

A possible way to deal with the various requirements of different applications is the use of search strategies for finding optimal or suboptimal solutions. The evolutionary computation techniques, such as Genetic Algorithms (GAs) and Variable Neighborhood Search (VNS) are examples of heuristic search strategies that can be used.

GAs [13] are search techniques that consider sets of candidate solutions (each solution is an *individual*, and the set is the *population*), which are varied according to two kinds of probabilistic rules: the *mutations*, which introduce perturbations into current solutions, producing new ones, and the *crossover*, which combine the information from previous solutions, producing new ones. Finally, the current population passes a *selection* procedure, that probabilistically increases the frequency of the best solutions in a new population, reducing the frequency of the worst ones. In recent years, it has been recognized that a key factor that determines the performance of GAs is the encoding employed for the representation of the solutions in the population. This is attributed to the fact that different encodings induce different neighborhoods, which lead to different behaviors of the variation mechanisms of mutation and crossover [5].

VNS techniques [16], on the other hand, usually evolve a single solution each time. This solution is subject to heuristic descent searches that find local minima in the attraction regions that are characterized by connected paths in a given neighborhood induced by an encoding. The heart of VNS techniques is the alternate usage of different encodings that induce different neighborhoods, which allows the algorithm to perform further descent steps after finding a local minimum in an encoding, by simply changing the encoding that is being used.

This paper deals with the problem of packet routing in MPLS systems. In the specific context of this problem, a new Multiobjective Genetic Algorithm, the VN-MGA (Variable Neighborhood Multiobjective Genetic Algorithm) is developed. The optimized routing tries to minimize the network cost and the amount of rejection of simultaneous requests, as well as perform a load balancing among routes. Using the proposed algorithm makes possible to deal with these conflicting QoS indicators, described as independent objective functions. Moreover, the set of solutions provides flexibility for the decision maker to select one or other goal according to the current state of the network.

The proposed VN-MGA is based on the classical NSGA-II [8] and has, as a distinctive feature, its crossover and mutation operators inspired in the concept of *variable neighborhood* of the VNS techniques. Two different encodings are employed: a low-level encoding, which encodes explicitly the routes that are followed by each requirement of service, and a high-level encoding, that encodes the permutations of the several requirements of service, defining the order in which they will be included in the solution. The crossover and mutation operators, acting in these two levels, are able to explore and to exploit the decision variable space with enhanced efficiency, leading to solutions that dominate the ones that appear in algorithm versions using only one level. It should be noticed that the proposed operators are problem-specific. In problems of combinatorial nature, it has been established that algorithms employing specific crossover and mutation operators can be much more efficient than general-purpose GAs [4].

A group of routing problems has focused on hybrid methods, since 2006 [19]. There are hybrid methods for the vehicle routing problem using Genetic Algorithms and Tabu Search [19] or combining VND (Variable Neighborhood Descent) and GRASP (Greedy Randomized Adaptive Search Procedure) [12] and also problems of another characteristics, such as pipeline petroleum distribution using GA and VNS [21]. However, those studies typically combine the different algorithms in a literal way, performing steps from one algorithm and from the other algorithm. The present authors have not identified any reference that performs an organic combination like the one proposed here.

This paper is organized as follows: Section II describes the problem and its modeling. Section III presents the VN-MGA. Section IV presents some results obtained with this approach and the section V concludes the paper.

2 Problem Description and Modeling

This study deals with the problem of choosing routes in a scenario of a corporative IP network with MPLS technology. The proposal is to minimize the network cost, to respond for the various user's requests ensuring the quality of service and to provide a load balancing between simultaneous streams. The network model is represented by the graph $G = (V, A)$, where V is the set of routers in the MPLS domain and $A = (i, j)$ is the set of links from node i to node j, or the links between the routers. The bandwidth of each link (i, j) is represented by B_{ij}. Each user request is represented by (o^k, d^k, b^k), where o^k and d^k indicate, respectively, routers of source and destination of traffic and b^k indicates the amount of bandwidth to be reserved for the request k. The set of requests is represented by R.

The objective functions are described by the equation (1), based on the work [20].

$$\min \begin{cases} F_1 = \sum_{k \in R} \sum_{(i,j) \in A} x_{ij}^k \\ F_2 = \sum_{k \in R} (1 - a^k) \\ F_3 = \alpha \end{cases} \tag{1}$$

s.t.

$$\sum_{(i,j) \in \Gamma_i^+} x_{ij}^k - \sum_{(l,i) \in \Gamma_i^-} x_{li}^k = \begin{cases} a^k & (o^k); \\ -a^k & (d^k); \\ 0, & \forall i \in V, \forall k \in R \end{cases} \tag{2}$$

$$\sum_{k \in R} b^k x_{ij}^k \leq \alpha B_{ij}, \forall (i, j) \in A \tag{3}$$

$$\sum_{k \in R} a^k \geq C \tag{4}$$

where

$$x_{ij}^k \in \{0, 1\}, \forall (i, j) \in A, \forall k \in R \tag{5}$$

$$a^k \in \{0, 1\}, \forall k \in R \tag{6}$$

$$\alpha \in [0, 1] \tag{7}$$

The objective function F_1 represents the sum of links used to accept a request k. The fewer links are used, the smaller is the delay for the data travel from origin to destination. F_2 aims to reduce the number of rejections of requests. The amount of rejection of the requests is related to the admission control of new connections, which determines if a connection can be admitted or not, according to the load network condition and the amount of bandwidth requested. The minimum number of requests that must be responded is represented by C, shown in Equation 4. In F_3, α represents (in relative terms) the load of the most used edge, with values varying from 0 to 1. Minimizing the amount of data traffic on the links means that the load is evenly distributed. Consequently, the network is balanced. The constraint 2 represents the classical flow conservation. In 3, the requested bandwidth (b^k) for a link (i, j) must be less than or equal to the available bandwidth.

The problem stated in equation (1) has several functions to be minimized, and therefore is a *multi-objective optimization* problem. A multi-objective optimization problem is defined as:

$$\begin{aligned} &\min \mathbf{f}(\mathbf{x}), \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_l(\mathbf{x})) \\ &\text{subject to: } \ \mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathcal{X} \end{aligned} \tag{8}$$

in which $\mathbf{x} \in \mathcal{X}$ is the *decision variable vector*, $\mathcal{X}$ is the *optimization parameter domain*, $\mathbf{f} \in \mathcal{F}$ is the *objective vector*, $\mathcal{F}$ is the *objective space*, in other words, $\mathcal{F} = \mathbf{f}(\mathcal{X})$.

The goal of some multi-objective optimization methods is to obtain estimates of the Pareto-optimal set [10], which contains the set of non dominated solutions of the multi-objective problem. A point $\mathbf{x}'$ is said to be dominated by another point $\mathbf{x}$ if the following relation holds:

$$\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}') \text{ and } \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}')$$

in which the relation operators $\leq$ and $\neq$ are defined as:

$$\mathbf{f}(\mathbf{a}) \leq \mathbf{f}(\mathbf{b}) \Leftrightarrow f_i(\mathbf{a}) \leq f_i(\mathbf{b}), \quad \forall i = 1, 2, \cdots, l$$

and

$$\mathbf{f}(\mathbf{a}) \neq \mathbf{f}(\mathbf{b}) \Leftrightarrow \exists i \in \{1, 2, \cdots, l\} : f_i(\mathbf{a}) \neq f_i(\mathbf{b})$$

in which **a** and **b** represent two different decision vectors.

In this way, the Pareto set $\mathcal{P}$ is defined as the set of non dominated solutions:

$$\mathcal{P} = \{\mathbf{x}^* | \nexists \mathbf{x} : \mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*) \ \wedge \ \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}^*)\} \,. \tag{9}$$

All solutions that are not dominated by any other decision vector of a given set are called *non dominated* regarding this set. A Pareto-optimal solution is a non dominated vector $\mathbf{x} \in \mathcal{X}$. The Pareto-optimal set of the multi-objective optimization problem is the set of all Pareto-optimal solutions. The image of this set in the objective space is called the *Pareto front* $\mathbf{f}(\mathcal{P})$.

3 Structure of Multiobjective Genetic Algorithm

The basic structure of the multiobjective genetic algorithm VN-MGA used here is the classical *Non-dominated Sorting GA* (NSGA-II), described in [8]. The following features of NSGA-II are used inside VN-MGA:

1. *Non-dominated sorting:* consists in sorting the solutions according to the non-dominance rank. An individual belonging to rank 1 is not dominated by any of the solutions, while an individual belonging to rank k is dominated by at least one individual that belongs to rank $k-1$. This ensures that solutions belonging to lower dominance ranks are better than solutions situated at higher ranks.
2. *Crowding-distance:* The crowding distance is used as a measure of occupation in the neighborhood of a solution in the objective space. The use of crowding distance helps to avoid situations where the obtained Pareto-set is too concentrated in a small portion of the Pareto set, leading the algorithm to more uniform samplings of the Pareto-set.
3. *Binary tournament:* consists in choosing two individuals randomly and comparing them according to some fitness function. The one with best fitness evaluation is selected.

We now depict the multiobjective optimization approach to the problem of optimizing routing in IP networks, capable of working with QoS parameters.

3.1 Variable Neighborhood Search

The Variable Neighborhood Search (VNS), proposed by [16], is a simple heuristic method which has the purpose of performing a global optimization using sequences of local searches. The main idea is to have a repertoire of operators which implicitly define two or more structures of neighborhood, and to switch between them. The search starts with a feasible solution and searches iteratively for a new one in the immediate neighborhood defined by the current search operators. By switching the operators, it is possible to change the neighborhood, which allows to perform descent searches in a new neighborhood. This allows to escape from points which represent local minima in some neighborhood, using the descent paths of other neighborhoods.

In literature, there are many variants of VNS which consider different sequences of neighborhoods, or different solution acceptance conditions. A basic version defines a set of neighborhoods $N = \left\{N^k, k = 1, ..., kmax\right\}$ and an initial solution x that will be used in the local search with the neighborhood N^1. This procedure is repeated for many iterations. The acceptance condition will choose between the previous local optimum and the new one. If this is the best solution, then the neighborhood for the next iteration will be in the first position. Otherwise, the neighborhood will follow the sequence. The algorithm proposed by [14] is described in Algorithm 1.

Based on distinct neighborhoods, this work proposes the integration of VNS concepts within Genetic Algorithms. During the evolution process, genetic operators of crossover and mutation are developed for each such neighborhood. In this way, the search in one neighborhood aids the search in the other one with alternated executions, exploring different search spaces.

Algorithm 1. Basic VNS

```
$k \leftarrow 1$
while $k \leq kmax$ do
    a) Perturbation: generate randomly a point $x'$ in $N^k(x)$;
    b) Local search: Local search from $x'$; Denote $x''$ as a local optimum obtained;
    c) Acceptance: if $f(x'') < f(x)$ then $x \leftarrow x''$ and continue the search on this neighborhood;
    else, $k \leftarrow k + 1$
end while
```

3.2 The Proposed Multiobjective Genetic Algorithm

The figure 1 illustrates a schema of the proposed algorithm, the VN-MGA. Crossover and mutation operations are defined at two levels. The number of generations is the criterion used to determine when to switch encodings. Thus, after N generations searching new routes at Level 1, the search for new solutions is made at Level 2.

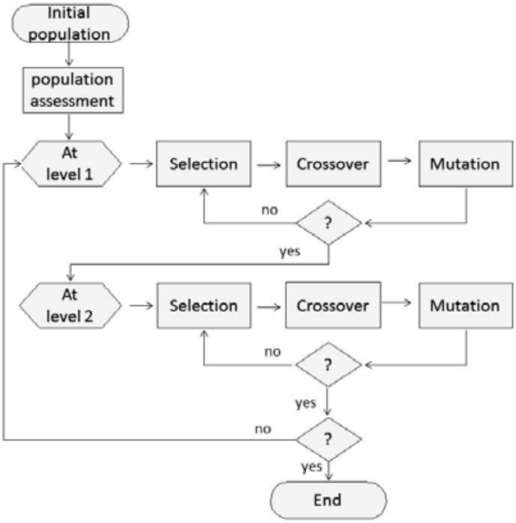


Fig. 1. Schema of the VN-MGA

3.3 Genetic Representation

The encoding was designed in two levels of operations. The Level 1 represents the codification of routing, i.e. the genetic operations focus on the sequences of arcs that form routes. The Level 2 encodes the sequence of requests, i.e. the sequence in which the requests should be included in the solution. The requests indicate the demand of N flows, given as origins and destinations. The two levels are considered alternately. Figure 2 illustrates the population at each level.

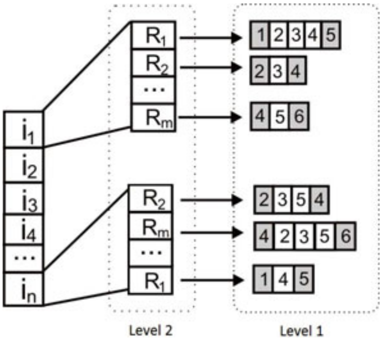


Fig. 2. Codification in two levels of operations

At Level 1, the individual is represented by a group of routes on each request, denoted by $i_1, i_2, \ldots, i_n$. Each path is described by a source, intermediates and destination nodes, represented by the node numbers, where the first one is the source node and the latter one is the destination node. Each request has a specific need for bandwidth according to its application, called Bandwidth Requested. Each link has a total Bandwidth Capacity. Thus, in order to respond to requests, the bandwidth requested must be less than or equal to the available one.

Considering that it is a multiobjective problem, not only the shortest path should be cogitated. For this reason, aiming to generate a diversity of individuals, the initial routes are generated from the Dijkstra's algorithm with random costs distributed on links, given the origins and the destinations.

The bandwidth is then withdrawn from the available one, representing the allocation of routes with bandwidth reservation. If the request cannot be met, i.e. the requested bandwidth is greater than the available one, then that request is rejected.

At Level 2, the individuals are represented by a set of requests, indicated by $R_1, R_2, \ldots, R_m$. As the algorithm does not differentiate the priority between simultaneous requests, the evaluation is made according to their position in the sequence. Thus, depending on the sequence, accepting or rejecting a request can impact the result of optimization.

3.4 Crossover Operators

In this study, two levels of crossover were proposed also. The first one consists of the exchange of genetic material between individuals at Level 1, which represents the routes. Two individuals are selected randomly, responding for the same request r of source O_r and destination D_r. If there is dominance between them, the dominant individual is selected. If there is no dominance, the crossover tries to join characteristics of both parents. If there is a node Nc in common between these individuals, the offspring is formed from node O_r up to Nc of the first parent i_1 and from Nc until Dr of the second parent i_2. Figure 3 illustrates the crossover process.

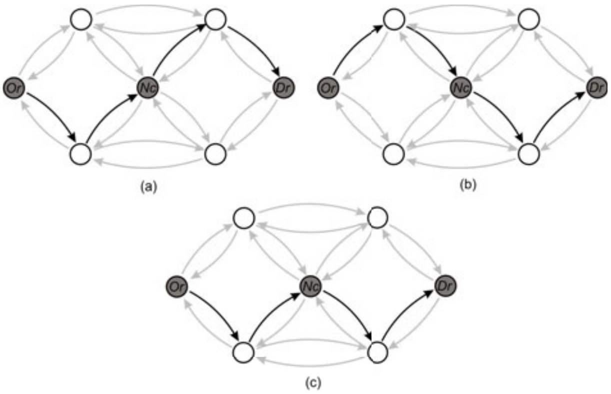


Fig. 3. Crossover with node in common [20]. (a) represents the route of an individual i_1 to a request r. (b) represents the route of an individual i_2 to a request r. (c) represents the route of the offspring to a request r.

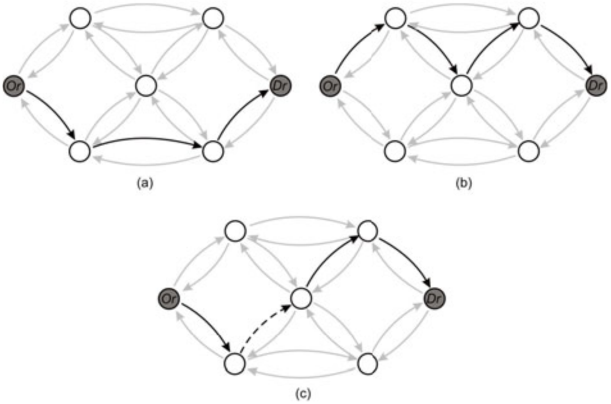


Fig. 4. Crossover without node in common [20]. (a) represents the route of an individual i_1 to a request r. (b) represents the route of an individual i_2 to a request r. (c) represents the route of the offspring to a request r.

If there is not a common node between individuals, the crossover attempts to find edges to link the paths. Thus, the offspring inherits the nodes from O_r until the new node interconnection of the first parent i_1 and from the new node until D_r of the second parent i_2. Figure 4 illustrates the crossover without a vertex in common.

If there is no node to link the paths, the offspring inherits one of the parents' path, randomly selected in order to compose the set of population.

In the Level 2 crossover, the individual is analyzed from the perspective of request sequence. Two routes representing the same request are randomly selected and swapped, generating a new combination of them, and therefore, a

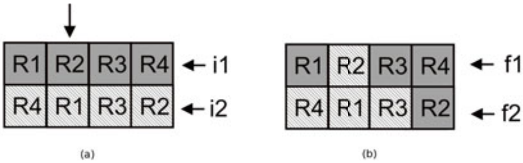


Fig. 5. Crossover at Level 2. (a) The request R_2 is selected in individual i_1. (b) The routes R_2 of i_1 and R_2 of i_2 are interchanged, generating the offspring.

new individual. Figure 5 illustrates this operation. The request R_2 is selected from the individual i_1. Then, looking for R_2 in i_2, the routes R_2 of i_1 and R_2 of i_2 are interchanged.

3.5 Mutation

The mutation operator is the responsible for the insertion of new genetic characteristics for individuals of the population. As for crossover, the mutation is defined for the two encoding levels, with Level 1 concerning the sequences of arcs that form the routes and Level 2 dealing with the sequences of requests, given pre-defined routes that were found with Level 1 operations.

For Level 1, the chromosome to be mutated and two cutoff points on that are chosen randomly. A new path, from the initial to the second cutoff point, is searched in the adjacency matrix. Therefore, a new section of the route is created. The search for a new sub-route is performed both in forward and backward directions (from the first to the second cutoff point, and in the opposite direction,

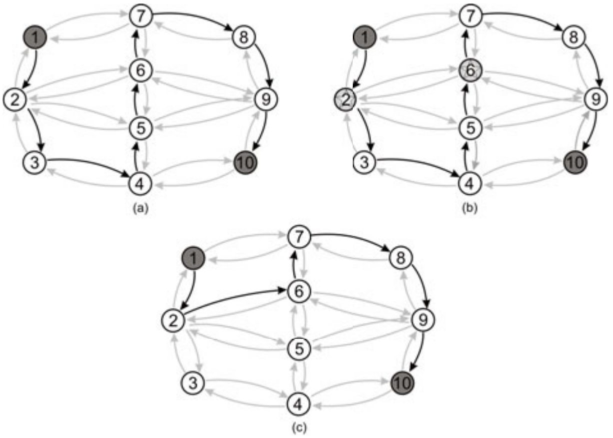


Fig. 6. Mutation at Level 1. (a) represents the route of an individual to a request r. (b) selection of the points 2 e 6 for a new sub-route. (c) Mutated individual.

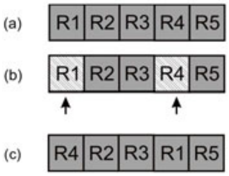


Fig. 7. Mutation at Level 2. (a) Represents the sequence of requests from individual. (b) The points selected for mutation. (c) Mutated individual.

from the second to the first cutoff point), alternately, avoiding any bias in this search. Figure 6 illustrates the process of mutation.

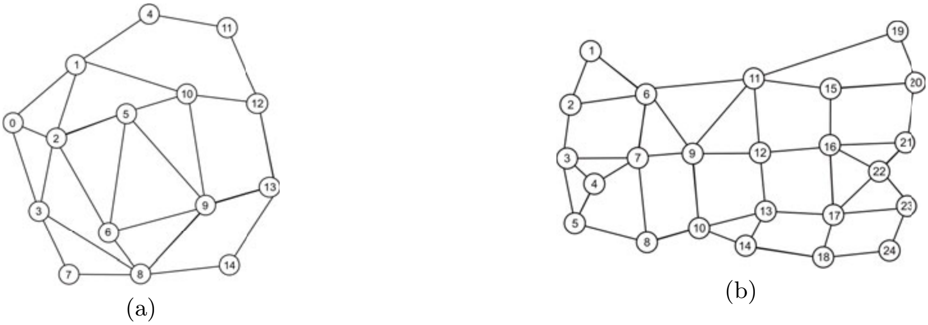
The mutation in Level 2 performs the permutation of requests. Similarly to the Level 1 mutation, at this stage two cutoff points are selected randomly, but now, considering the requests. Then, a swap between these two requests changes the sequence of individuals. Figure 7 represents the mutation at Level 2.

4 Results

Tests for the proposed algorithm, VN-MGA, have been performed with some network instances, commonly used for telecommunications problems, such as Dora (Fig. 8(a)) and Carrier (Fig. 8(b)).

Although these network models are standard, there is no other reported work that uses the same scenario assumed here. For this reason, there is not a reference value, except the approximation of the functions described by [20], but in a different scenario. These approximated results are represented by circles in Figures 9 and 10 and indicate the Level 1.

In the same figures, the crosses represent Level 2, i.e., the results are obtained considering the codification of requests. The asterisks indicate the combination of both levels 1 and 2, representing the VN-MGA. This means that, after searching



(a)

(b)

Fig. 8. (a) Dora. (b) Carrier.

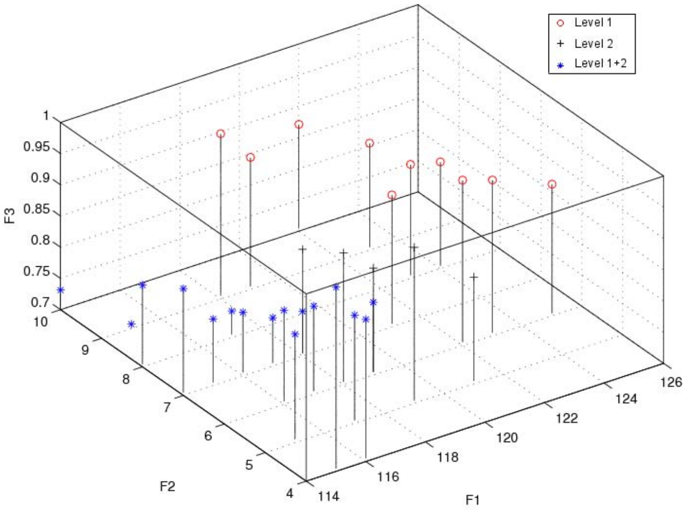


Fig. 9. Results for Dora topology

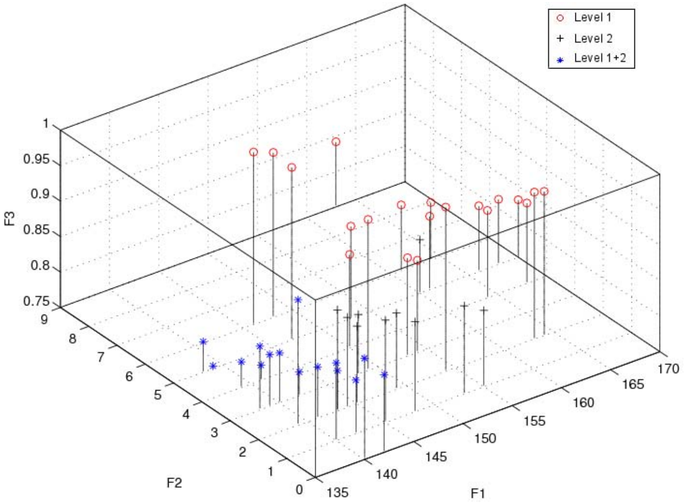


Fig. 10. Results for Carrier topology

for good routes, the interchange of requests is performed in order to reorganize their sequence and after a reorganization, new routes are searched again, and so on.

As shown, the variable encoding described by two levels, improves the final quality of the routing. It can be observed that the solutions delivered by VN-MGA dominate the solutions obtained by single-level searches.

The parameters used for the experiment are described by Table [1].

Table 1. Parameters for the algorithm

Mutation Probability	0,4
Crossover Probability	0,9
Available Bandwidth for each link	1024Kbps
Requested Bandwidth	200Kbps e 400Kbps
Number of generations	50
Number of individuals	50
Number of requests	50

5 Conclusions and Future Work

This paper proposed an algorithm to optimize multiple objectives that represent Quality of Service indices on IP networks. The proposed algorithm, VN-MGA, is a Genetic Algorithm based on the NSGA-II, with the particular feature that each solution has two different encodings, at Level 1 and Level 2. At Level 1, the solution is encoded considering as decision variables the arcs that form the routes to be followed by each request. At Level 2, the solution is encoded with the routes considered as fixed, and the sequence of requests considered as the decision variable. The results suggest that local minima can be indeed avoided using this approach.

There are future works to be conducted in two distinct directions: the specific problem of packet routing in MPLS, and the theoretical problem of employing variable neighborhoods (or different solution encodings) in generic evolutionary algorithms.

Concerning the MPLS problem, a challenging area concerns a quantitative analysis, covering sensitivity and scalability. The sensitivity deals with fault tolerance in paths or routers and the capacity of re-routing of the proposed method. Using new scenarios, it is possible to assess the scalability in order to quantify the gain that is expected with the application of the proposed algorithm. Within this perspective, it is also possible to suggest new models for telecommunications networks. In any case, the proposed approach delivers a reasonable diversity of solutions belonging to the Pareto Front. So, it offers a larger range of options for the decision maker in different situations, such as in: (i) network congestion that occur in rush moments, or (ii) using applications that require a small delay, or (iii) responding to concurrent requests that do not present stringent requirements of delay, but require large bandwidths, among others.

Concerning the theoretical problem of studying variable neighborhoods in generic evolutionary algorithms, there are several open issues. The authors intend to tackle, in the near future, some issues related to the usage of encodings that allow metric operations [17,5]. It should also be noticed that an analogous of the ideas presented here has been also developed in our research group for the case of continuous variables [6].

References

1. Alvarado, C., Herazo, I., Ardila, C., Donoso, Y.: Aplicación de NSGA-II y SPEA-II para la optimización multiobjetivo de redes multicast. Ingeniería y Desarrollo 17, 28–53 (2005)
2. Andrade, A.V.: Provisionamento de Qualidade de Serviço em Redes MPLS utilizando Algoritmos Bio-inspirados em um Ambiente de Trfego Auto-Similar. Ph.D. thesis, UFMG (December 2008)
3. Awduche, D., Chiu, A., Networks, C., Elwalid, A.: Overview and Principles of Internet Traffic Engineering. Request for Comments 3272 (May 2002)
4. Carrano, E.G., Soares, L.A.E., Takahashi, R.H.C., Saldanha, R.R., Neto, O.M.: Electric distribution network multiobjective design using a problem-specific genetic algorithm. IEEE Trans. Power Delivery 21(2), 995–1005 (2006)
5. Carrano, E.G., Takahashi, R.H.C., Fonseca, C.M., Neto, O.M.: Non-linear network topology optimization - An embedding vector space approach. IEEE Transactions on Evolutionary Computation 14, 206–226 (2010)
6. Carrano, E.G., Moreira, L.A., Takahashi, R.H.C.: A new memory based variable-length encoding genetic algorithm for multiobjective optimization. In: Proceedings of the 6th Int. Conf. Evolutionary Multicriterion Optimization, EMO 2011 (2011) (submitted)
7. De Giovanni, L., Della Croce, F., Tadei, R.: On the impact of the solution representation for the Internet Protocol Network Design Problem with max-hop constraints. Networks 44(2), 73–83 (2004)
8. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. Evol. Comput. 2(6), 182–197 (2002)
9. Dias, R.A.: Engenharia de Trafego em Redes IP sobre Tecnologia MPLS: Otimizaçao Baseada em Heurısticas. Ph.D. thesis, UFSC (2004)
10. Ehrgott, M.: Multicriteria Optimization. Springer, Heidelberg (2000)
11. Erbas, S., Erbas, C.: A multiobjective off-line routing model for MPLS networks. In: Providing Quality of Service in Heterogeneous Environments-Proc. of the 18th International Teletraffic Congress (ITC-18), vol. 16, pp. 471–480. Citeseer (2003)
12. de Freitas, L.M.B., Montané, F.A.T.: Metaheurísticas vns-vnd e grasp-vnd para problemas de roteamento de veículos com coleta e entrega simultânea. In: XI Simpósio de Pesquisa Operacional e Logística da Marinha (2008)
13. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Boston (1989)
14. Hansen, P., Mladenovi, N.: Variable Neighborhood Search: Principles and applications. European Journal of Operational Research 130(3), 449–467 (2001)
15. Maia, N.A., de Errico, L., Caminhas, W.M.: Combining MPLS, Computational Intelligence, and Autonomic Computing into a Self-Managing Traffic Engineering System.. In: Proceedings of the Second Latin American Autonomic Computing Symposium LAACS 2007, pp. 1–6 (2007)
16. Mladenovi, N., Hansen, P.: Variable neighborhood search. Computers & Operations Research 24(11), 1097–1100 (1997)
17. Moraglio, A., Kim, Y.-H., Yoon, Y., Moon, B.-R.: Geometric crossovers for multiway graph partitioning. Evolutionary Computation 15(4), 445–474 (2007)

18. Paul, P., Raghavan, S.: Survey of QoS routing. In: Proceedings of the International Conference on Computer Communication, vol. 15, p. 50. Citeseer (2002)
19. Perboli, G., Pezzella, F., Tadei, R.: EVE-OPT: a hybrid algorithm for the capacitated vehicle routing problem. Mathematical Methods of Operations Research 68(2), 361–382 (2008)
20. Santos, F.A.: Otimização Multiobjetivo aplicada a alocação dinâmica de rotas em redes de telecomunicações. Master's thesis, UFMG (March 2009)
21. de Souza Filho, E.M.: Variable Neighborhood Search (VNS) aplicado ao problema de distribuição dutoviária. Master's thesis, UFRJ - Universidade Federal do Rio de Janeiro (2007)
22. Wang, Z., Crowcroft, J.: Quality-of-service routing for supporting multimedia applications. IEEE Journal on Selected areas in communications 14(7), 1228–1234 (1996)

Real-Time Estimation of Optical Flow Based on Optimized Haar Wavelet Features

Jan Salmen[1], Lukas Caup[1], and Christian Igel[2]

[1] Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany
{Jan.Salmen,Lukas.Caup}@ini.rub.de
[2] Department of Computer Science, University of Copenhagen, 2100 Copenhagen Ø, Denmark
igel@diku.dk

Abstract. Estimation of optical flow is required in many computer vision applications. These applications often have to deal with strict time constraints. Therefore, flow algorithms with both high accuracy and computational efficiency are desirable. Accordingly, designing such a flow algorithm involves multi-objective optimization. In this work, we build on a popular algorithm developed for real-time applications. It is originally based on the Census transform and benefits from this encoding for table-based matching and tracking of interest points. We propose to use the more universal Haar wavelet features instead of the Census transform within the same framework. The resulting approach is more flexible, in particular it allows for sub-pixel accuracy. For comparison with the original method and another baseline algorithm, we considered both popular benchmark datasets as well as a long synthetic video sequence. We employed evolutionary multi-objective optimization to tune the algorithms. This allows to compare the different approaches in a systematic and unbiased way. Our results show that the overall performance of our method is significantly higher compared to the reference implementation.

1 Introduction

Optical flow can be defined as the "the distribution of apparent velocities of movement of brightness patterns in an image" [1], which result from the projection of a 3D scene onto an image plane. Estimating optical flow is common to many computer vision applications since the resulting flow field is very valuable, amongst others, for detection and tracking of objects, deriving structure from motion, estimation of ego-motion, and collision avoidance. These problems often arise in the context of real-world applications (e.g., robotics, surveillance, automotive) where hard time constraints have to be met. Therefore, there is a special interest in flow algorithms that offer good accuracy while being computationally efficient.

In this study, we propose to employ evolutionary multi-objective optimization for tuning flow algorithms with respect to these two partially conflicting objectives. This allows analysis of different possible trade-off solutions and therefore systematic comparisons.

We consider a *feature-based* algorithm [2], which was designed for driver assistance systems. This approach combines computational efficiency, high accuracy, robustness,

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 448–461, 2011.

and ease of implementation. The Census transform [3] is used for the description of small image patches. This allows for table-based matching and tracking of interest points over time, resulting in a sparse flow vector field. In this work, we show how to modify the algorithm in order to make it more flexible and powerful. We suggest to replace the Census transform by Haar wavelet features, which are state-of-the-art for real-time computer vision.

This article is organized as follows. In the next section, we give an overview over related work. In section 3, the reference algorithm is introduced. Section 4 presents our approach. Multi-objective optimization is introduced in section 5. Our experiments are described in section 6, finally the results are discussed.

2 Related Work

For a detailed overview and comparison of many different flow algorithms, we refer to Barron et al. [4] and Baker et al. [5] and references therein. Here, we focus only on approaches closely related to our method, namely real-time capable and feature-based flow algorithms.

Lucas and Kanade [6] proposed an iterative method that was adapted by Bouguet [7] for pyramid based feature tracking. His implementation is publicly available in the *OpenCV* library[1]. We consider this algorithm as a baseline for our experiments. Tomasi and Kanade [8] proposed tracking of features obtained based on eigenvalues of small image patches. Optical flow algorithms based on this principle are fast and accurate [9]. Similar results can also be obtained with correlation-based approaches [10], or block-matching algorithms [11]. Stein proposed the algorithm that serves as starting point for our method [2]. It uses the Census transform to generate signatures for image patches that can be used like *hash values* to handle interesting points. This allows for very efficient temporal analysis. This approach is described in more detail in the next section.

The local, feature-based approaches mentioned above calculate sparse flow vector fields. While these algorithms are able to handle large displacements, they may produce noisy results depending on local structure. It is essential to attach great importance to temporal integration and evaluation of pixel neighborhoods.

Dense optical flow can be obtained based on the variational method proposed by Horn and Schunck [1]. Implementations of this basic technique are likely to fail for large displacements and for small moving objects because of the smoothness constraints. Moreover, such implementations typically are not well suited for real-time applications. However, different improvements and extensions have been proposed recently which address these issues [12,13,14,15].

3 Original Method

The Census transform as proposed by Zabih and Woodfill [3] is based on the 3×3 neighborhood $\mathcal{P}$ of a pixel. We denote the center pixel by P_0 and the eight ones

[1] `http://sourceforge.net/projects/opencvlibrary`

surrounding it by $P_1, \ldots, P_8$. The signature $\xi(\mathcal{P})$ is calculated *bitwise*, where the i-th digit ξ_i is set to 0 or 1 depending on the intensities of P_i:

$$\xi_i(P_0, P_i) = \begin{cases} 0, & P_i > P_0 \\ 1, & P_i \leq P_0 \end{cases} . \tag{1}$$

This is extended in [2] in order to distinguish "similar" pixels by introducing a new parameter ϵ and computing the signature as

$$\xi_i(P_0, P_i) = \begin{cases} 0, & P_0 - P_i > \epsilon \\ 1, & |P_0 - P_i| \leq \epsilon \\ 2, & P_i - P_0 > \epsilon \end{cases} . \tag{2}$$

Additionally, Stein proposed to consider larger neighborhoods and therefore not only pixels directly adjacent to the center. For his experiments, he used signatures with up to 20 digits.

This signature is computed for every pixel in a camera image. The positions of all image patches with signature s in frame t are stored in a table $T_t(s)$. Flow hypotheses between frames $t-1$ and t are computed by considering all pairs of entries from $T_{t-1}(s)$ and $T_t(s)$ for all relevant signatures s. In this step, two restrictions are considered: The lengths of flow vectors can be limited and the relative brightness difference between the center pixels is restricted to avoid bad flow hypothesis. All resulting hypotheses in frame t are stored in a list H_t.

To increase speed and robustness of the approach, not all possible signatures have to be processed. Stein proposed to use a so-called black-list to exclude signatures that code for infeasible image patches (e.g., homogeneous image regions or regions related to the aperture problem). Furthermore, lists table entries $T_t(s)$ with more than mdp (*maximum discriminative power*) elements are ignored for the calculation of flow hypothesis because the number of possible hypothesis increases exponentially.

The hypotheses calculated between two frames are generally not unique and therefore not reliable. In order to compute final flow vectors for frame t, longer-term temporal analysis is performed based on H_{t-1} and H_t. For each hypotheses h in frame t, it is checked if there was another hypothesis from frame $t-1$ with similar length and similar orientation ending approximately where the new vector starts. Figure 1 illustrates this analysis. Finally, reliable flow vectors are obtained from current hypotheses which have more than p_{min} predecessors.

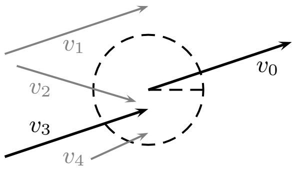


Fig. 1. Temporal analysis in frame t. $v_1, \ldots v_4$ from H_{t-1} are possible predecessors for v_0 from H_t. The vector v_1 is not valid because it is outside the search radius, v_2 because of the angular difference to v_0, and v_4 because of the relative length difference. v_3 is the only valid predecessor.

4 Proposed Algorithm

In this section, we present our new flow algorithm. It is based on the one described in the previous section but is more flexible. In particular, we show that it can optionally allow for sub-pixel accuracy.

4.1 Haar Features for Index-Based Matching

Haar wavelet features are state-of-the-art for real-time computer vision. Their popularity is mainly based upon the efficient computation using the *integral image* proposed by Viola and Jones [16]. Haar wavelet features were successfully applied in many computer vision applications, especially for object detection, classification, and tracking [16,17,18,19]. Figure 2 shows examples of six basic types of Haar Wavelet features. Their responses can be calculated with 6 to 9 look-ups in the integral image, independently of their absolute sizes.

It stands to reason to consider Haar wavelet features also for the calculation of optical flow. They could offer a corporate preprocessing for even more different applications and therefore serve as a universal basis for real-time computer vision systems.

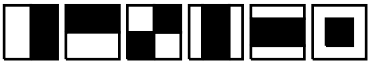


Fig. 2. Basic types of Haar wavelet features

In the original flow algorithm, a feature-based index is obtained for every pixel through the Census transform. We calculate a similar value based on responses R of a bunch of different Haar wavelet features centered on that pixel. The signature for a pixel is calculated as

$$\xi_i(R) = \begin{cases} 0, & R_i < -\epsilon_i \\ 1, & |R_i| \leq \epsilon_i \\ 2, & R_i > \epsilon_i \end{cases}, \tag{3}$$

where ϵ_i now are individual thresholds used for discretization of the continuous feature responses. We further extend this by allowing for more than one ϵ per feature. For the example of two thresholds, $\epsilon_i^{(0)}$ and $\epsilon_i^{(1)}$, the discretization is

$$\xi_i(R) = \begin{cases} 0, & |R_i| \leq \epsilon_i^{(0)} \\ 1, & \epsilon_i^{(1)} > R_i > \epsilon_i^{(0)} \\ 2, & R_i > \epsilon_i^{(1)} \\ 3, & -\epsilon_i^{(1)} < R_i < -\epsilon_i^{(0)} \\ 4, & R_i < -\epsilon_i^{(1)} \end{cases}. \tag{4}$$

The temporal analysis (i.e., interest point matching and tracking) can be done analogously to the original approach. The Census-based method is embedded in our approach as a special case, because the definition of Haar-like wavelets allows for regions with size of a single pixel. That is, the calculations for the Census transform can be modeled with Haar wavelet features. Of course, in this case, calculation of the integral image would not lead to any gain in speed.

4.2 Optional Sub-pixel Refinement

Using Haar wavelet features allows to perform flow computation with sub-pixel accuracy by employing an optional refinement step: The position of the end points of all flow hypothesis in frame t are refined in a way that the feature responses are most similar to the responses at the start point in frame $t - 1$.

We propose to perform *bilinear interpolation* in the radius of one pixel around the original (pixel-accurate) end point in order to find the best matching position. The resolution between pixels can be chosen arbitrarily. For our experiments, we used steps of one-tenth pixel. To calculate matching costs c for two feature vectors R and S with n components each we used the *sum of absolute differences* $c\,(R, S) = |R_1 - S_1| + |R_2 - S_2| + \ldots + |R_n - S_n|$.

The proposed sub-pixel refinement requires that the raw feature responses are either stored for start points of all correspondence hypothesis or that they are calculated anew only for final flow hypothesis. It depends on the problem at hand which of these two solutions is preferable in terms of efficiency. Here, we recalculate feature responses when needed for refinement.

5 Multi-objective Optimization

In our experiments, we evaluate the performances of various approaches on different datasets. For a fair and meaningful comparison, all algorithms must be setup “as good as possible”. Adapting parameters in such cases is typically done manually in a more or less systematic way. We propose to use rely on evolutionary optimization instead. It allows to do a more extensive search in an impartial manner.

We give a short overview over evolutionary multi-objective optimization with an emphasis on sorting candidate solutions in section 5.1. The variation of candidate solutions is described in section 5.2. Of course, results obtained by automatic optimization have to be watched as critically as results obtained by manual tuning. We discuss these issues in section 6.3.

5.1 Evolutionary Multi-objective Optimization

Optimizing parameters of sparse optical flow algorithms is not straightforward as their performance cannot be quantified by a single measure: maximization of the number of flow vectors and maximization of their mean accuracy are two *conflicting goals*. Therefore, we suggest to perform *multi-objective optimization* (MOO, *vector optimization*) [20]. The goal of vector optimization is to find a diverse set of solutions that approximates the set of *Pareto-optimal* trade-offs. A solution is Pareto-optimal if it is *not dominated* by another solution. In our application a solution is not dominated if on the one hand no other solution exists with more flow vectors and at least as good accuracy and on the other hand each solution with better accuracy does not have more flow vectors.

We consider evolutionary MOO. The crucial step in this process is selection. Here, we adopt the selection procedure from the MO-CMA-ES [21] for comparing and sorting candidate solutions to determine the parents for the next generation.

For the problem at hand, the vector valued quality of a solution g is given by $\boldsymbol{\Phi}(g) = (n, 1/e)$, where n is the mean number of flow vectors and e is the mean endpoint error as proposed by [5]. Both objectives are to be maximized. A solution g (weakly) dominates another solution g' iff it is better in one of the two objectives and is not worse in the second.

Based on this concept, each candidate solution g_i in a set $M = \{g_0, \ldots, g_n\}$ can be assigned a rank $r(g_i)$ that expresses its level of non-dominance. The set M is partitioned into subsets $M_1, M_2, \ldots, M_{n_{\text{levels}}}$. Candidate solutions in M_1 are not dominated by any other solution in M, solutions in M_2 are only dominated by members of M_1, solutions in M_3 are only dominated by solutions in $M_1 \cup M_2$, and so on. The rank $r(g)$ of a candidate solution g is defined as the index of the set containing it.

A second criterion is needed in order to sort solutions having the same rank index r. We do this based on their *contributing hypervolume* as proposed in [21,22,23]. The dominated hypervolume $\mathcal{V}(g)$ of a candidate solution g is defined as the volume of all points in the objective space (here $\mathbb{R}^2_{\geq 0}$) that are dominated by $\boldsymbol{\Phi}(g)$. The dominated hypervolume of a set M_i is defined as the volume of the union of the dominated hypervolumes of all points in M_i. The contributing hypervolume $\mathcal{V}^{\text{cont}}_{M_i}(g)$ for a candidate solution $g \in M_i$ is given by the share of the total dominated hypervolume that is lost if the solution is removed: $\mathcal{V}^{\text{cont}}_{M_i}(g) = |\mathcal{V}(M_i) - \mathcal{V}(M_i \setminus \{g\})|$. Figure 3 illustrates this concept.

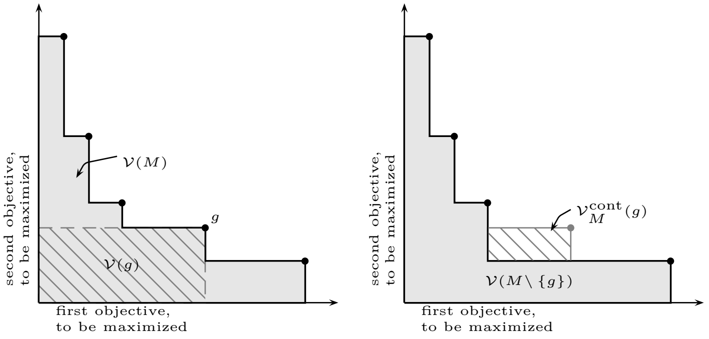


Fig. 3. The concepts of hypervolume and contributing hypervolume. Left: Dominated hypervolume of solution g and dominated hypervolume of set M. Right: Contributing hypervolume of $g \in M$ and dominated hypervolume of set $M \setminus \{g\}$.

A set M^* only containing non-dominated solutions is called a *Pareto set*, the corresponding vectors $\{\boldsymbol{\Phi}(g) \,|\, g \in M^*\}$ a *Pareto front*. After optimization, M^* represents different possible trade-offs. This does not only give insights into the characteristics of the problem at hand, but can be useful in practice, for instance, to adapt the behavior of an application online.

5.2 Mutating solutions

In our optimization problem, we allow the number of features, their individual parameters, and the flow algorithm parameters to vary. This calls for a variable length representation. We use an encoding and variation operators resembling those proposed in [18]. In the following, we briefly outline the tailored mutation procedure using the example of a Haar feature based algorithm.

Each time when mutating an individual, tossing a coin decides whether the parameters of the flow algorithm are varied or the feature set. The flow algorithm parameters comprise mdp, d_{int}, l_{max}, r, d_{ang}, d_{l}, and p_{min} (see Table 1). They are mutated using standard operators.

If the feature set is mutated, it is decided whether the number of features is changed or the individual features are varied. The former happens with a probability of $p_{\text{size}} = 0.20$. Features are then added or removed with equal probability. New features are created with a random type (see Figure 2) and width and height each between 2 and 12.

Alternatively, with a probability of $1 - p_{\text{size}} = 0.80$, we mutate the features. Each Haar feature has real-valued, integer, and nominal parameters. The mutation operators *changeSize*, *changeType*, and *mutateEpsilons* are applied to each Haar feature with a probability of $p_{\text{mut}} = 0.50$ each. In *changeSize*, the width and height of the feature are changed independently by $+1$ or -1. In *changeType*, a new random basic type is assigned to the feature. In *mutateEpsilons*, the thresholds $\epsilon_i^{(0)}, \epsilon_i^{(1)}, \ldots$ are mutated. This also includes mutating the number of thresholds used. We allowed up to three of these thresholds per feature.

A repair mechanism ensures that only feasible solutions are generated: For each parameter, we defined a minimal and maximal value according to their meaning. For example, it is not reasonable to have negative values for differences (e.g., d_{int} and r) or Haar features smaller than 2×2 pixels.

6 Experiments

We compare the original Census based and our Haar wavelet based approach in two different scenarios. As a baseline for comparison, we also evaluate the performance of the pyramid based Lucas and Kanade algorithm [7] using *good features to track* [9] from *OpenCV*. The setup of our experiments is described in section 6.1, the results in section 6.2. We discuss these results as well as crucial points of our experiments in section 6.3.

6.1 Setup

The first comparison is performed on Middlebury benchmark datasets[2]. The Middlebury benchmark [5] was designed to evaluate only dense flow algorithms. However, there are some additional datasets available that we can use for evaluation: *RubberWhale*, *Hydrangea*, *Grove2*, *Grove3*, *Urban2*, and *Urban3*. All these sets consist of eight frames and evaluation is performed only for the last frame in each case. We use a common parameter set for all sequences. Figure 4 shows three example frames out of the datasets.

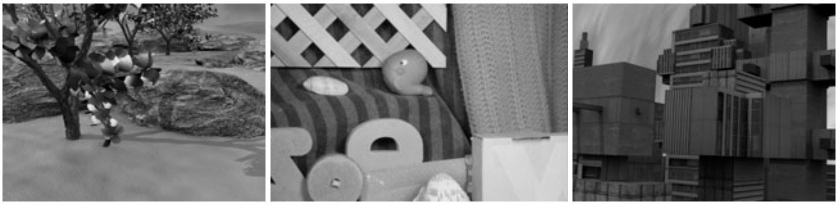


Fig. 4. Frames out of the Middlebury datasets considered for experiments

The second comparison is performed on an *Enpeda* benchmark sequence[3]. Long synthetic video sequences have recently been introduced by Vaudrey et al. [24]. We consider the second sequence which has complex textures and hard shadows. Optimization is performed on frames 80 to 180. The subsequent frames are used after optimization to test how the optimized solutions generalize. Figure 5 shows three example frames out of the sequence.

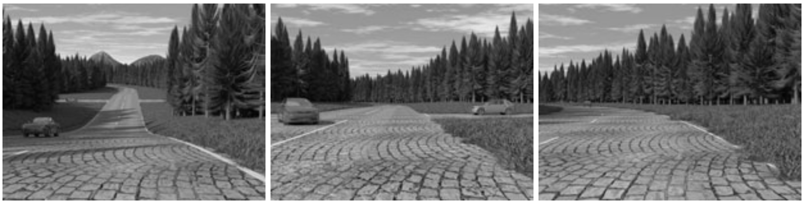


Fig. 5. Frames out of the synthetic Enpeda sequence considered for experiments

For the Middlebury as well as the Enpeda benchmark, we perform MOO for all three approaches. For the *OpenCV* algorithm, there are 8 parameters to be optimized. For the Census based as well as the Haar feature based approach, we optimize the features used and the algorithm parameters. As the temporal analysis is the same in both methods, many parameters that are optimized are identical, see Table 1. For the Census based

[2] `http://vision.middlebury.edu/flow`

[3] `http://www.mi.auckland.ac.nz/index.php?option=com_content&view=article&id=52`

Table 1. Parameters of the flow algorithm used

Description		unit
Maximum discriminative power	mdp	
Maximum intensity difference	d_{int}	[%]
Maximum vector length	l_{max}	[pixel]
Radius for predecessors	r	[pixel]
Maximum angular difference	d_{ang}	[°]
Maximum length difference	d_{l}	[%]
Minimal number of predecessors	p_{min}	

approach, we used a setup with 12 digits and therefore additionally optimized individual thresholds for each digit. For the new approach based on Haar features, we additionally optimized the thresholds $\epsilon_i^{(0)}, \epsilon_i^{(1)}, \ldots$, types, and sizes of features as described in section 5.2.

For the Census based and the Haar feature based approach, we do not use a blacklist as proposed in the original work. The setup of such a list would make statistical analysis necessary, which would be performed after optimization and only improve the results. As this is the same for both methods, no bias is introduced in our comparison.

As framework for the evolutionary multi-objective optimization, we use the *Shark* open-source machine learning library [25], which is freely available[4]. It provides implementations of efficient algorithms for computing the (contributing) hypervolume used for selection.

We performed 5 independent optimization runs for each experiment. In every optimization run, we considered 1000 generations. We used elitist $(\mu+\lambda)$-selection. In each generation, $\lambda = 15$ new candidate solutions result from recombination and mutation of $\mu = 15$ parent solutions. We made sure that all parameters remain within a reasonable domain. Solutions generated during optimization which are not feasible regarding our application, namely solutions having less than 500 flow vectors or mean endpoint error larger 5 pixels, are discarded. We selected the best solutions (according to the definitions from Sec. 5.1) from all runs in order to obtain final sets for comparison. For further details, we refer to the source code and its documentation which is made publicly available.

6.2 Results

The resulting sets of optimized solutions for the Middlebury datasets are shown in Figure 6. They visualize different trade-offs between Φ_1, the mean number of flow vectors, and $1/\Phi_2$, the mean endpoint error.

Using the original Census based approach, it was possible to find up to $22,286$ flow vectors per frame (with mean endpoint error of 4.95). Using the new Haar feature based approach, it was possible to find up to $46,999$ flow vectors per frame (with mean endpoint error of 2.89). Sub-pixel refinement allowed to further increase the accuracy of

[4] `http://shark-project.sourceforge.net`

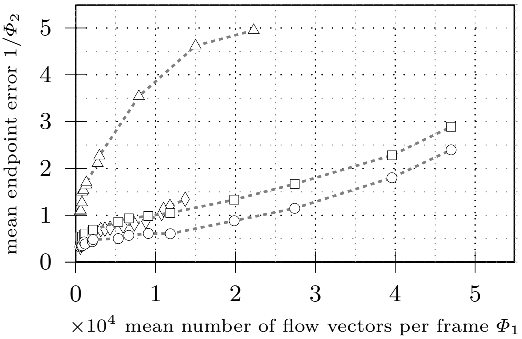


Fig. 6. Results of optimization for Middlebury datasets: Pareto fronts for Census based approach (triangles), Haar feature based approach with pixel (squares) and sub-pixel (circles) accuracy, and *OpenCV* algorithm (diamonds)

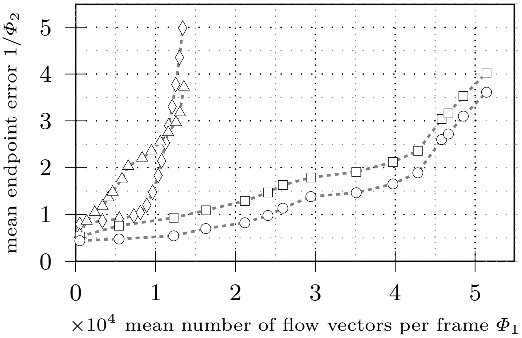


Fig. 7. Results of optimization for Enpeda sequence: Pareto fronts for Census based approach (triangles), Haar feature based approach with pixel (squares) and sub-pixel (circles) accuracy, and *OpenCV* algorithm (diamonds)

our approach. The *OpenCV* approaches allowed to find up to $13,669$ flow vectors per frame (with mean endpoint error of 1.46).

Figure 7 shows the Pareto fronts after optimization for the Enpeda sequence. The *OpenCV* algorithm allowed to find up to $12,415$ flow vectors. For the Census- based approach the maximum number of flow vectors found was $13,545$ and for the Haar feature based approach $51,434$. Sub-pixel refinement improved the mean endpoint error in our approach by 0.4 pixels on average. Optimized Haar wavelet features for two different solutions are shown in Figure 8. Table 2 shows the optimized parameter settings for different trade-offs for both approaches.

The generalization performances of the solutions on the rest of the sequence are shown in Figure 9. The Census based approach and the *OpenCV* algorithm performed significantly better in the test than in the training. Therefore one can conclude that

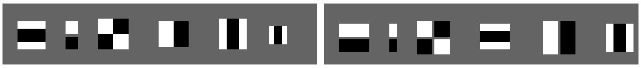


Fig. 8. Optimized feature sets from two different solutions. The features have sizes of (a) 11×11, 5×11, 12×12, 12×10, 11×12, and 7×7. (b) 12×11, 3×11, 13×13, 12×10, 13×13, and 11×11.

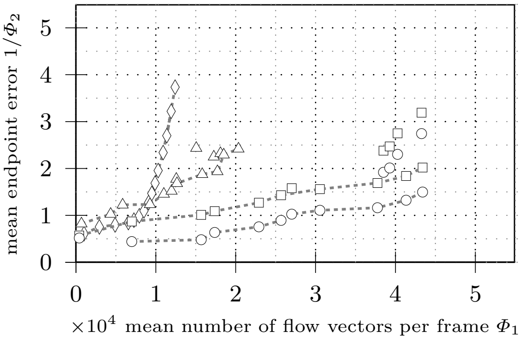


Fig. 9. Generalization results of optimized solutions on the rest of the Enpeda sequence: Census based approach (triangles), Haar feature based approach with pixel (squares) and sub-pixel (circles) accuracy, and *OpenCV* algorithm (diamonds)

Table 2. Optimized parameters from different trade-off solutions for the Enpeda sequence

Φ_1	$1/\Phi_2$	mdp	d_{int}	$l_{\max}$	r	d_{ang}	d_{l}	$p_{\min}$
Census based								
502	0.57	33	1.2	43	0.8	0.32	0.11	2
6569	2.03	70	0.5	40	5	0.05	0.1	1
13545	3.72	70	1	45	5	0.23	0.1	1
Haar feature based								
521	0.53	33	0.6	80	0.6	0.23	0.1	3
12264	0.93	70	1.6	40	1.6	0.05	0.15	3
25940	1.63	70	2	40	5	0.05	0.1	2
39709	2.12	70	2	40	3.3	0.05	0.1	1
51435	4.03	69	2	46	4.6	0.15	0.74	1

this latter part of the sequence is “easier”. The Haar feature based approach performed slightly better for solutions up to 4×10^4 flow vectors and slightly worse for solutions with more flow vectors.

The optimized solutions for the Census based approach result in runtimes between 200 ms and 400 ms on a standard desktop PC with 2.2 GHz. The solutions of the Haar wavelet based approach result in runtimes between 250 ms and 600 ms. It is not surprising that these runtimes are comparable as the implementation of temporal analysis is identical for both approaches. The optimized solutions for the *OpenCV* algorithm resulted in runtimes between 200 ms and 1100 ms.

A typical result of the new Haar feature based flow algorithm is shown in Figure 10, where an optimized solution was considered that combines very small mean endpoint errors and an adequate number of flow vectors. This setup is suitable for

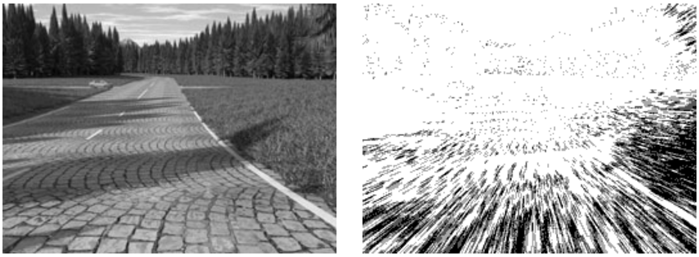


Fig. 10. Camera image from the Enpeda sequence (left) and resulting flow field (right), final flow vectors shown in black

real-world applications in the context of driver assistance systems, for example, for object segmentation and tracking, collision avoidance, or ego-motion estimation.

6.3 Discussion

In our experiments, the three algorithms were optimized in the same way for the same scenarios. The Census based and the new Haar feature based approach share most of their parameters because the temporal analysis is the same. Therefore the significant differences in performance shown in our experiments solely result from the different feature basis. The Haar feature based approach is more powerful in general: The mean endpoint errors were the smallest for every number of flow vectors in all experiments, especially when the optional sub-pixel refinement was performed. Our method additionally allowed to have a significantly higher number of flow vectors compared to the other approaches, in particular almost the fivefold amount in the Enpeda benchmark.

One could argue that the Haar feature based approach has the most degrees of freedom (as feature types and sizes can be adapted) and therefore profits most from optimization. While this might be true, it is exactly this flexibility that makes the approach particularly attractive. Despite this flexibility, most of the optimized solutions showed good generalization, there was no *over-fitting*.

The execution speed was not an objective in the optimization performed here, therefore the solutions found lead to comparatively slow execution speed, still most of them meet real-time constraints. All optimized solutions for the new method use 6 Haar features which was the maximal allowed number in the optimization. Nevertheless, the algorithm can generally be used in real-time applications requiring higher frame rates, especially as the table-based matching and the temporal analysis are suited very well for implementation on embedded hardware.

Evaluation of different optimized solutions also allows for statements regarding on how the parameters for the proposed flow algorithm should be set. Most of the optimized features are relatively large, 85 % in the final solutions have widths and heights larger than 10 pixels. This shows that considering relatively large neighborhoods is preferable in the given framework. Using Haar features, this is possible without any drawbacks regarding computational cost. This is an advantage over other filters. Most parameters are correlated to the trade-off between high number of flow vectors and high accuracy in some degree. Nevertheless, dependencies are manifold and the interplay is complex. Therefore, manual adjustment (e.g., for a fixed desired trade-off) is challenging and evolutionary optimization has shown to be a valuable tool.

Appropriate parameters are problem class specific. They clearly have to be different for an automotive application (with many relatively large displacements) and for entrance hall surveillance. Therefore the parameters found here are well suited for the scenarios considered but have to be changed for other contexts. This can easily be done for the proposed method.

7 Conclusion

Many computer vision applications benefit from fast and accurate estimation of optical flow. We considered a popular feature-based method which relies on the Census transform and table-based matching. We proposed to employ Haar wavelet features within this framework in order to make it more flexible and powerful.

Haar wavelet features are state-of-the art for real-time computer vision, particularly for object detection and classification. It is highly desirable to use them in more tasks within the same application to benefit from a shared preprocessing. We showed how Haar wavelet features can be applied for computation of optical flow after appropriate discretization. The resulting approach is versatile and allows for solutions that satisfy different trade-off requirements.

The performance of the new method was systematically assessed and compared to the original approach considering two different benchmarks. This was possible by evolutionary multi-objective optimization, which provides the opportunity to handle the two conflicting goals *high number of flow vectors* and *high accuracy*. In our experiments, the new approach significantly outperformed the original one. For any bound on the maximum mean error, the method can find a number of flow vectors several times larger compared to the original approach.

Our experiments suggest how to set up the new flow algorithm in general. In particular, considering relatively large neighborhoods has shown to be most successful. This favors Haar wavelets features, because the computational cost of calculating them is independent of their sizes.

References

1. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA (1980)
2. Stein, F.: Efficient computation of optical flow using the census transform. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 79–86. Springer, Heidelberg (2004)
3. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 151–158. Springer, Heidelberg (1994)
4. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. International Journal of Computer Vision 12(1), 43–77 (1994)

5. Baker, A., Scharstein, D., Lewis, J., Roth, S., Blak, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1–8. IEEE Press, Los Alamitos (2007)
6. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
7. Bouguet, J.Y.: Pyramid implementation of the Lucas Kanade feature tracker. In: OpenCV Documentation, Intel Corporation Microprocessor Research Labs (2000)
8. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University (1991)
9. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600. IEEE Press, Los Alamitos (1994)
10. Camus, T.: Real-time quantized optical flow. Journal of Real-Time Imaging 3, 71–86 (1995)
11. Zhang, D., Lu, G.: An edge and color oriented optical flow estimation using block matching. In: Proceedings of the IEEE International Conference on Signal Processing, vol. 2, pp. 1026–1032. IEEE Press, Los Alamitos (2000)
12. Bruhn, A., Weickert, J., Kohlberger, T., Schnörr, C.: A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. International Journal of Computer Vision 70(3), 257–277 (2006)
13. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L^1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)
14. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 41–48. IEEE Press, Los Alamitos (2009)
15. Steinbruecker, F., Pock, T., Cremers, D.: Large displacement optical flow computation without warping. In: Proceesings of the IEEE International Conference on Computer Vision, pp. 1609–1614. IEEE Press, Los Alamitos (2009)
16. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision 57(2), 137–154 (2001)
17. Papageorgiou, C., Poggio, T.: A trainable system for object detection. International of Journal Computer Vision 38(1), 15–33 (2000)
18. Salmen, J., Suttorp, T., Edelbrunner, J., Igel, C.: Evolutionary optimization of wavelet feature sets for real-time pedestrian classification. In: Proceedings of the IEEE Conference on Hybrid Intelligent Systems, pp. 222–227. IEEE Press, Los Alamitos (2007)
19. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 260–267. IEEE Press, Los Alamitos (2006)
20. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Dordrecht (1999)
21. Igel, C., Hansen, N., Roth, S.: Covariance matrix adaptation for multi-objective optimization. Evolutionary Computation 15(1), 1–28 (2007)
22. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653–1669 (2007)
23. Suttorp, T., Hansen, N., Igel, C.: Efficient covariance matrix update for variable metric evolution strategies. Machine Learning 75(2), 167–197 (2009)
24. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: Proceedings of the Conference on Image and Vision Computing New Zealand, pp. 1–6. IEEE Press, Los Alamitos (2008)
25. Igel, C., Glasmachers, T., Heidrich-Meisner, V.: Shark. Journal of Machine Learning Research 9, 993–996 (2008)

Multi-objective Genetic Algorithm Evaluation in Feature Selection

Newton Spolaôr[1,2], Ana Carolina Lorena[1], and Huei Diana Lee[2]

[1] Grupo Interdisciplinar em Mineração de Dados e Aplicações/
Universidade Federal do ABC
Santo André, Brasil
[2] Laboratório de Bioinformática/Universidade Estadual do Oeste do Paraná
Foz do Iguaçu, Brasil
{newtonspolaor,aclorena,hueidianalee}@gmail.com

Abstract. Feature Selection may be viewed as a search for optimal feature subsets considering one or more importance criteria. This search may be performed with Multi-objective Genetic Algorithms. In this work, we present an application of these algorithms for combining different filter approach criteria, which rely on general characteristics of the data, as feature-class correlation, to perform the search for subsets of features. We conducted experiments on public data sets and the results show the potential of this proposal when compared to mono-objective genetic algorithms and two popular filter algorithms.

Keywords: filter feature selection, feature importance measures, multi-objective genetic algorithms.

1 Introduction

Enormous volume of data has been collected, due to the development of technology, and organized in Databases (DB). Computational processes like Data Mining (DM) may be applied in order to analyze these DB. DM enables the construction of logical hypothesis (models) from data, potentially extracting useful knowledge for specialists, that can be used as a second opinion in decision making processes [18].

The DM process is mainly composed of pre-processing, pattern extraction and pos-processing. Pre-processing involves the proper representation of the data into forms like attribute-value, in which lines and columns represent, respectively, examples and features (attributes, characteristics) of the data set. Other pre-processing tasks include cleaning the data and Feature Selection (FS), which is the focus of this work. The pattern extraction phase involves the construction of models from data, using, for example, Machine Learning algorithms. The obtained models are evaluated and consolidated by the specialists at the end of the process (pos-processing).

FS may be formulated as a search for an optimal subset of features in a DB, in which each state of the search space represents a possible subset of features [25].

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 462–476, 2011.

The optimality of this subset may be estimated according to a maximization or minimization function of one or more measures (criteria) of importance of features. Applying FS allows mapping the original data to a projection in which the examples are described by part of the features. This leads to a dimensional reduction of the data set. Models constructed using these projections may have lower complexity and potentially superior or equivalent quality when compared to those generated using the original data. In addition, FS may help on a better comprehension of the domain, by maintaining only the features with a good ability, according to some importance criterion, to describe the inherent patterns within the data and helps to reduce the effects of the curse of dimensionality [25].

Searching related to FS is usually a combinatorial process [7], precluding the investigation of all subsets. This is one of the motivations to apply heuristic search methods such as Genetic Algorithms (GA) [27] in this process. Furthermore, it may be interesting to find subsets of features that optimize different importance criteria, leading to the motivation of using Multi-objective Optimization strategies (MO) [6]. In the literature, there are many applications of Multi-objective Genetic Algorithms (MOGA) in different areas and tasks, including FS [36,5,13,44,37,3,19,42].

The objective of this work is to evaluate the application of MOGA to FS based on different filter importance criteria. The performed experiments investigate distinct combinations of these criteria, what is not performed in works related to the filter approach [5,37,3,42]. This work also differentiates from previous work [35,36,5,29,33,42,34] by including a comparative evaluation of the MOGA with distinct mono-objective GA, where each GA optimizes one filter importance criterion individually, and two popular filter FS algorithms. The selected subsets are evaluated through the construction of models using two pattern extraction algorithms in nine benchmark data sets. The predictive ability of these models is statistically compared to the performance of models built using all features.

This study is part of the Intelligent Data Analysis project (IDA) [35,36,23], which is developed in a partnership among the “Universidade Federal do ABC” (UFABC), the “Laboratório de Bioinformática/Universidade Estadual do Oeste do Paraná” (LABI/UNIOESTE), the “Laboratório de Inteligência Computacional/Universidade de São Paulo” (LABIC/USP) and the “Serviço de Coloproctologia/Universidade Estadual de Campinas” (UNICAMP).

This work is organized as follows: in Section 2 concepts related to FS and also importance measures used in the MOGA are described. The complete proposal is described in Section 3 and its evaluation, using nine data sets from a public repository, is presented in Section 4. Final considerations are made in Section 5.

2 Feature Selection

Feature Selection may be viewed as a dimensional reduction process of a data set in order to maintain only its most important features according to some criterion. The importance criteria are usually based on the principles of relevance and non-redundancy among features [15,23,17] and may be organized into measures of consistency, dependency, distance, information and precision [25]. In this work, one measure of each of the mentioned categories was used with the MOGA, excluding precision. These measures were chosen from work related to filter Feature Selection [1,42,20,38,28].

All considered data sets are for supervised learning and related to classification problems. In these data sets, each example (or case) has an associated label (or class) and the objective is to construct predictive models, which are capable of predicting the label of new cases previously unknown. The data set is composed of n pairs $(\mathbf{x}_i, y_i)$, in which $\mathbf{x}_i = (x_i\,(1)\,, \ldots, x_i\,(m))$ represents an example with m features and y_i corresponds to its class. The exclusion of a relevant feature F_1 results in a worse predictive performance of the correspondent classification model. Two features F_2 and F_3 are said to be non-redundant when they are not significantly correlated.

It is relevant to mention that the feature importance estimation may be considered in two ways: individually or in subsets. Nevertheless, the individual evaluation methods are incapable of removing redundant features, as they may present the same relevance [17]. For this reason, in this work we considered FS in subsets. Individual importance evaluations are combined into a unique resultant value, representing the subset of features as a whole.

The importance of features may be viewed according to the interaction with the pattern extraction algorithm [25]. In the wrapper approach, a pattern extraction algorithm, which will be used later for the construction of models, is considered for selecting features. For each subset, a model using this specific algorithm is constructed and evaluated. In the filter approach, feature subsets are evaluated before the pattern extraction step, and considers general characteristics of the data, such as statistical measures, to select the important features. A third approach is the embedded, in which the process of selecting features is performed internally by the pattern extraction algorithm, as in the case of decision trees [31].

Importance Measures. Importance measures inspired in the concept of consistency value, for example, chooses subsets of features that minimize the occurrence of inconsistent pairs of examples in discretized data sets, that is, which present identical values in each feature but different labels [1]. The Inconsistent Example Pairs (IP) measure identifies the inconsistency rate by the ration of the number of inconsistent pairs of examples and the total number of pairs of examples.

Correlation measures enable a redundancy analysis of the data set when estimating the prediction capability of a feature. The Attribute Class Correlation (AC) [38] exemplifies this category, and is described by Equation 1, where w_i will be 1 if i is selected and 0 otherwise; $\phi\,(.,.) = 1$ if j_1 and j_2 have distinct labels or -0.05 otherwise. $|.|$ denotes the module function. The formulation of $C\,(i)$ demonstrates that this measure highlights feature values that show the most distinct values for examples of different classes.

$$AC = \left(\sum w_i C\left(i\right)\right) / \left(\sum w_i\right) \tag{1}$$
$$\text{where } C\left(i\right) = \frac{\sum_{j_1 \neq j_2} \left|x_{j1}\left(i\right) - x_{j2}\left(i\right)\right| \phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}\right)}{n\left(n-1\right)/2}.$$

The Inter-Class Distance measure (IE) [42] estimates the existent separability between classes when the set of examples is described only by the investigated subset of features. The separability maximization may be useful to generate classification models, as the differentiation of diverse patterns is favored. Equation 2 presents IE, where $\mathbf{p}$ is the central example (centroid) of a data set with k classes, $d(.,.)$ denotes the Euclidean distance, and $\mathbf{p}_r$ and n_r represent, respectively, the central example and the number of examples in class r.

$$IE = \frac{1}{n} \sum_{r=1}^{k} n_r d\left(\mathbf{p}_r, \mathbf{p}\right). \tag{2}$$

The Laplacian Score (LS) [20] is also based on distance and is inspired by the possibility of identifying examples with affinity when they are relatively next to each other. In classification, for example, this behavior is potentially observed among instances of the same label, highlighting the importance of modeling the related local structure. Herewith, LS proposes building a nearest neighbor graph, in which each node corresponds to a distinct example and the nearest examples are connected by arcs. The S weight matrix of this graph is considered in Equation 3, with $\mathbf{x}\left(i\right) = \left[x_1\left(i\right), x_2\left(i\right), \cdots, x_n\left(i\right)\right]^T$ and $\mathbf{1} = \left[1, \cdots, 1\right]^T$. This formula includes the matrices $D = diag\left(S\mathbf{1}\right)$, in which $diag(.)$ extracts the diagonal matrix, and the Laplacian Graph [8] $L = D - S$.

$$LS\left(i\right) = \frac{\tilde{\mathbf{x}}\left(i\right)^T L \tilde{\mathbf{x}}\left(i\right)}{\tilde{\mathbf{x}}\left(i\right)^T D \tilde{\mathbf{x}}\left(i\right)} \tag{3}$$
$$\text{where } \tilde{\mathbf{x}}\left(i\right) = \mathbf{x}\left(i\right) - \frac{\mathbf{x}\left(i\right)^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1}.$$

Information based measures may be applied to reduce the uncertainty associated to the investigated problem. Representation Entropy (RE) [28], for example, enables the investigation of the information distribution among features and, consequently, to estimate the involved redundancy [41]. RE is presented by Equation 4, in which the λ_i eigenvalues are extracted from a covariance matrix of features of m order.

$$RE = -\sum \tilde{\lambda}_i \log \tilde{\lambda}_i \tag{4}$$
$$\text{where } \tilde{\lambda}_i = \frac{\lambda_i}{\sum \lambda_i}.$$

Precision measures consider information like the accuracy rate of the model in the classification of examples described by a subset of features or other estimate of the models' quality. Usually these measures are related to the wrapper approach and are not considered in this work.

3 MOGA in Feature Selection

MOGA offers the combination of GA and MO for the solution of search and optimization problems with multiple objectives [9]. Searching for subsets of important features in a data set can be considered a multi-objective task, since there are multiple criteria for measuring their importance, and each one of them considers different aspects of data [1,42,20,38,28].

With the goal of ranking risk factors related to premature births, in [42] the Non-dominated Sorting Genetic Algorithm (NSGA-II) [12] MOGA was used in FS relating, through the Pareto strategy, importance measures IE, AC and Intra-Class Distance. Some results are superior to those of models built using all features and also of other techniques for FS.

In [5] the NSGA-II algorithm is applied in supervised and semi-supervised FS in data sets of hyperspectral images. Two measures were optimized simultaneously: discrimination between classes and spatial invariance of the features. In general, the results obtained demonstrate a superior performance of MOGA over mono-objective GA.

The same MOGA is used in [3] for FS in the classification of microarray gene expression profiles. Because of the nature of these data, which generally have few examples and many features, the classification task becomes more complex, motivating FS. The importance measures of cardinality and ability to discriminate examples were jointly optimized. Experimentally, there were accuracy gains when compared to a mono-objective GA and other techniques for FS.

The inter and intra correlation measures proposed by [37] were applied for FS in data sets for the analysis of credit risk, using the algorithm NSGA-II. Experimentally, the model built using the features selected by the MOGA had a better performance than those models generated using all features and also using features selected by mono-objective GA and the Relief technique [40].

This work differs from previous work by investigating some combinations of filter importance measures belonging to different categories. The individuals were encoded using a binary chromosome with m genes, each of which corresponds to a distinct feature. A gene i with value 1 represents the selection of its respective feature, while the value 0 indicates its exclusion. A randomly initialized population of individuals is then evolved until a number of generations is reached. The NSGA-II MOGA was used, as in the previous related work.

The importance measures used as objective functions to be optimized are those discussed in Section 2, which belong to the classes: consistency, dependence, distance and information. The aim is to exploit complementarities between representatives of measures from different categories. We investigated the optimization of these measures in pairs always involving IE and some other measure. This choice is based on previous results presented in [35,36], where the combinations involving the IE measure were more successful experimentally.

Experiments with three objectives led to a greater computational cost and little gains in other aspects, such as in the reduction obtained on the subsets of selected features. Furthermore, it is known that MOGA based on the Pareto theory do not scale well for optimization problems with more than three objectives [21]. For these reasons, only pairs of importance measures are considered in this work.

LS is the only measure used in the study which evaluates the importance of each feature individually. In its case, we used the average value calculated for each selected feature (with value 1 on a chromosome s). The other measures are calculated for each chromosome, using the subset of features represented by genes with value 1.

We used the one-point crossover, bit-flip mutation and binary tournament [27] in the MOGA. NSGA-II returns a set of optimal solutions, representing different tradeoffs between the objectives considered. We used the Compromise Programming (CP) technique [43] to select a single solution from this set due to its relative simplicity.

4 Experimental Evaluation

We applied the NSGA-II for FS described in the previous section in nine data sets from the UCI repository[1] [2]: Australian (A), Dermatology (D), Ionosphere (I), Lung cancer (L), Sonar (S), Soybean small (Y), Vehicle (V), Wine (W) and Wisconsin breast cancer (B). All features in these data sets are numerical and have continuous or discrete values. Table 1 presents, for each data set, the Majority Class Error (MCE) rate, which corresponds to the error rate obtained by classifying all data in the majority class, and the number (♯) of examples, features and classes.

Table 1. Data sets information

	A	**D**	**I**	**L**	**S**	**Y**	**V**	**B**	**W**
♯Examples	690	358	351	32	208	47	846	569	178
♯Features	14	34	34	56	60	35	18	30	13
♯Classes	2	6	2	3	2	4	4	2	3
MCE	44.49	68.99	35.9	59.37	46.63	63.83	74.23	37.26	60.11

We used the NSGA-II implementation available in the Platform and programming language Independent interface for Search Algorithms (PISA) [4], with the following parameters: $\alpha = 50$, $\mu = 50$, $\lambda = 50$, $crossover\ rate = 0.8$, $mutation rate = 0.01$, $stopping criterion = 50 generations$. The parameters α, μ and λ correspond, respectively, to the population size and the number of parents and children individuals after reproduction. Their values were defined based on related work. Another tool used in the implementations was the GNU Scientific Library (GSL)[2], which enables the implementation of the covariance matrices associated with the RE measure.

[1] Supported by the Turing Institute in Glasgow (Vehicle).

[2] http://www.gnu.org/software/gsl

As previously mentioned, we have investigated multi-objective combinations involving the IE measure and each of the other four importance measures described in Section 2, resulting in four distinct multi-objective settings. The evaluation of the subsets of features selected by the MOGA in each multi-objective setting was performed by building classification models using projections of the data sets containing the selected features. Classification algorithms J48, an implementation of C4.5 [31], and Support Vector Machines (SVM) [10], from the Weka tool [40], were used in the induction of classifiers. Their parameter values were kept default. These classifiers were selected due to: the relatively low number of parameters, in the case of J48; and robustness to high dimensional data, in the case of SVM.

We also implemented five mono-objective GA for FS, each of them using one of the importance measures discussed in this work as fitness function. The same binary encoding and genetic operators from MOGA were used. The results of the FS algorithms Correlation-based Feature Subset Selection (CFS) [16] and Consistency Subset Eval (CSE) [24] from literature are also presented. CFS chooses subsets of features highly correlated with the class and that have low inter-correlation, while CSE analyzes the consistency of the data projections obtained using different subsets of features.

CFS and CSE filter algorithms come from the Weka tool and were employed with default parameter values. The mono-objective GA's parameters number of generations, population size and probabilities of crossover and mutation were changed in order to be identical to those used for NSGA-II. The population size was set to 50 and the seed of GA, as for NSGA-II, was set randomly for each run. The classification models defined using all the features in each data set were included as baselines (c_a). In LS we used as neighborhood of each example its five nearest neighbors in terms of distance.

In the experiments, each data set d was initially divided according to Stratified Cross-Validation (SCV) into 10 folds, which leads to 10 pairs of training and test sets. Because of MOGA stochasticity, it was executed five times for each training partition f_i, and for each multi-objective setting m_s. One unique subset of features is identified for each MOGA run, using CP. This results in five subsets of features per multi-objective setting. This enables the generation of five different projections of the data partition f_i. After training classification models using the five projections and evaluating them on their corresponding test partitions, 50 accuracy rates are obtained. Similarly, we counted up the Percent of Reduction (PR) in the amount of original features for each run. The mean values of these evaluation measures are reported for each setting m_s. The mono-objective GA are subjected to a similar procedure, while the other FS algorithms and the baselines are evaluated by the mean values obtained in a unique run for each of the 10 folds of d.

4.1 Results

For each data set, we show in Figures 1 and 2 the PR and the accuracy rate of the J48 and SVM models for each FS algorithm when related to those rates

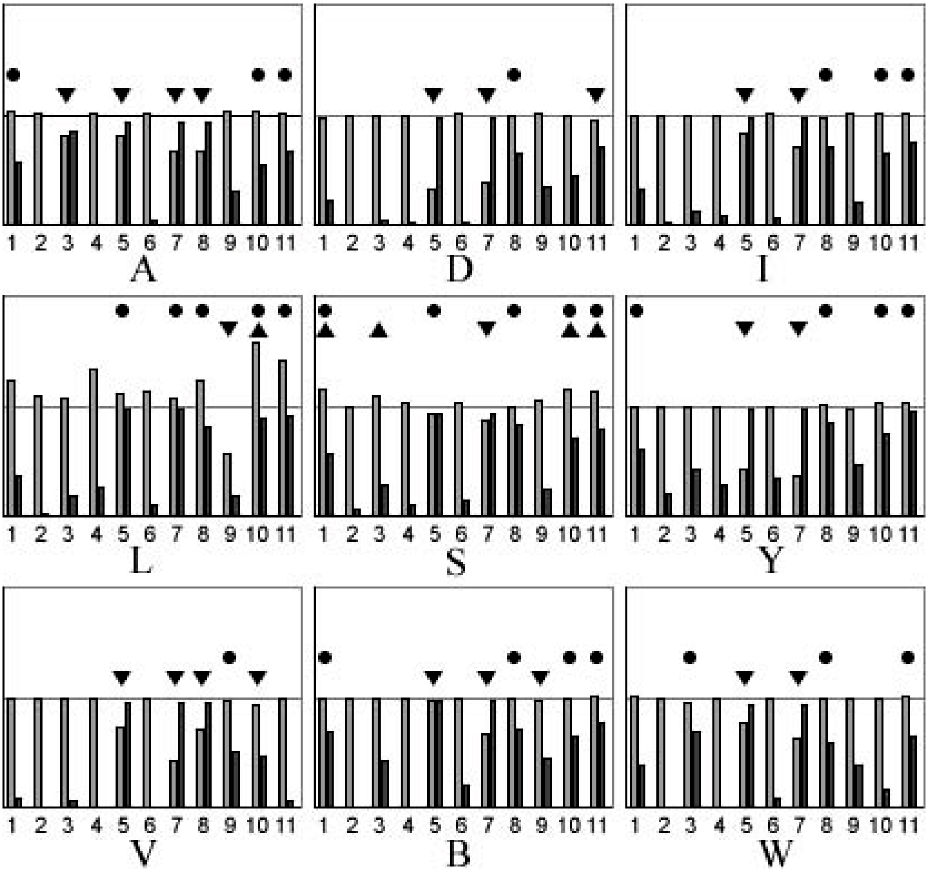


Fig. 1. J48 models generated after applying: (1) IE + AC, (2) IE + IP, (3) IE + LS, (4) IE + RE, (5) AC, (6) IE, (7) LS, (8) IP, (9) RE, (10) CFS and (11) CSE

obtained when using all features (c_a). Therefore, if a classifier c_i and c_a accuracy rates are, respectively, 85.36% and 83.48%, the graph displays for c_i the result of the ratio between these rates (1.02). The horizontal line corresponds to the point where this ratio reaches the value 1 in each graph, that is, when the rates are equal for both c_i and c_a. The dark bars represent PR and the light bars show the accuracy rates of the classifiers built using the features selected by each FS algorithm. Therefore, if the top of a light bar is above the baseline line, the accuracy of the model represented is higher than that of the baseline. The black bars will never exceed the baseline line, because PR is always less than 100% of the original number of features per data set. An aggressive PR is identified when the top of its bar is close to the baseline line.

We noticed that, in general, the magnitude of the standard deviations of the accuracy rates for each c_i had no strong discrepancy to those achieved by c_a.

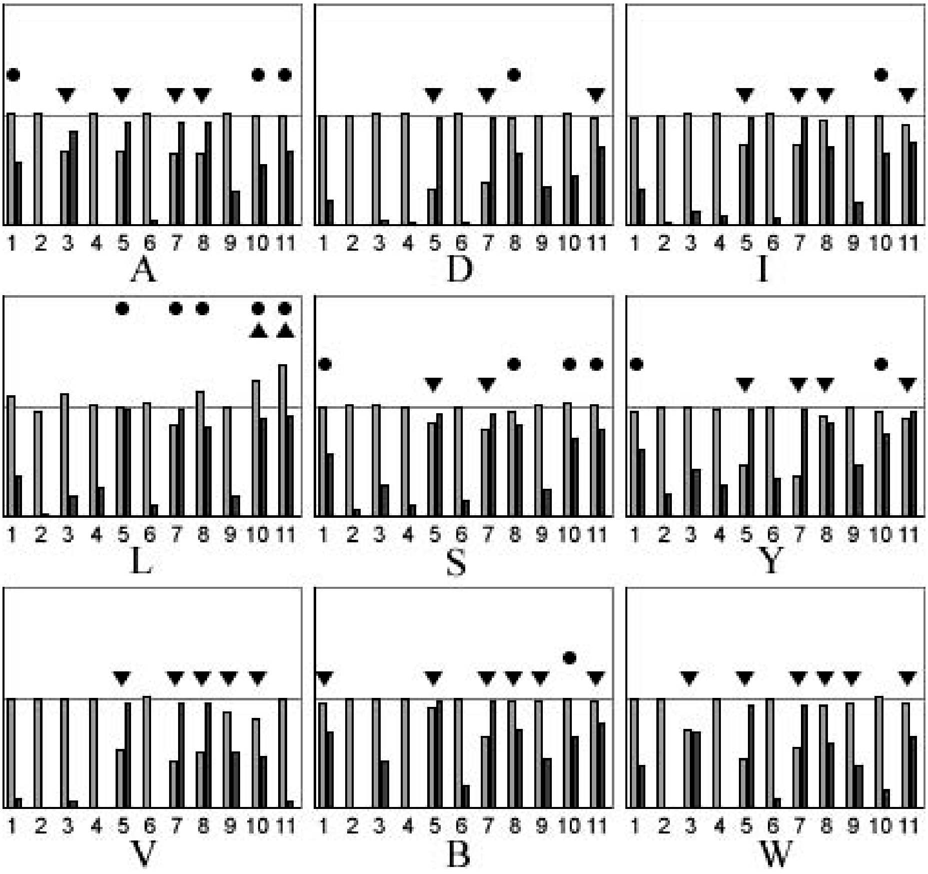


Fig. 2. SVM models gerarated after applying: (1) IE + AC, (2) IE + IP, (3) IE + LS, (4) IE + RE, (5) AC, (6) IE, (7) LS, (8) IP, (9) RE, (10) CFS and (11) CSE

Importantly, the FS embedded in J48 was not investigated in this work, therefore all PR illustrated refer to subsets of features identified by the FS algorithms evaluated.

Since we do not have assurance of normality, we employed the non-parametric Kruskall-Wallis test [22] separately for each data set to identify statistical differences at 95% of significance level between each of the algorithms evaluated and the baseline in terms of accuracy rate. Using a unique baseline implies in less statistical comparisons, softening the multiplicity effect [32]. Models with higher and lower statistical accuracy performance when compared to c_a are highlighted in the graphs, respectively, with a triangle pointing up and down. Models that had accuracy not statistically lower than that of c_a, while providing PR greater than 50%, are highlighted with a circle. Table 2 summarizes these information, presenting the total number of models significantly better (in brackets) and with

Table 2. Models not statistically inferior to c_a

	IE + AC	IE + IP	IE + LS	IE + RE	AC	IE
J48	8 (1)	9 (0)	7 (1)	9 (0)	2 (0)	9 (0)
SVM	8 (0)	9 (0)	7 (0)	9 (0)	1 (0)	9 (0)
PR	42.07 (20)	3.18 (6.7)	33.84 (28.5)	8.53 (11.7)	95.17 (2.1)	10.94 (10.7)

	LS	IP	RE	CFS	CSE
J48	1 (0)	7 (0)	7 (0)	6 (2)	7 (1)
SVM	1 (0)	3 (0)	6 (0)	7 (1)	3 (1)
PR	95.24 (1.9)	78.22 (12.3)	33.82 (11.7)	58.11 (21)	69.41 (25.7)

no significant difference when compared to c_a, for each classifier. It also shows the mean PR and standard deviation (in parentheses) by FS algorithm for all data sets.

4.2 Discussion

We identified 7 models (3.5% of the total) with significant superiority and 135 models (68.2%) with no significant difference of results when compared to the baseline, of which 46 had PR higher than 50%. A reduction in the number of features with the maintenance or improvement in accuracy when compared to the baseline is important, because it allows reducing the computational cost of the classification model induced and can also contribute to improve its comprehensibility [25,38]. However, most of these occurrences are concentrated in experiments related to the multi-objective setting IE + AC and to the CFS and CSE algorithms. This behavior is reinforced by the results shown in Table 2, where these algorithms showed PR greater than 40% and are associated to all cases of statistical superiority when compared to the baseline. We also indentified in several experiments, mainly those related to mono-objective GA using measures AC and LS, that a too aggressive dimensional reduction led to a predictive performance statistically inferior to that of c_a.

The MOGA based on the IE + AC measures stood out in comparison to the other settings by allowing the generation of models with lower computational complexity and predictive performance statistically similar to the baseline in different data sets. Additionally, in the Sonar data set it was possible to obtain a J48 model with superior performance when compared to c_a. These results reinforce previous experimental evidence [35,36] of the importance of selecting features that maximize the separability between classes for supervised classification problems. The combined use of IE with other measures of importance can contribute, for example, to select just one feature from two or more features that are equivalent in terms of separability.

The mix between IE and AC explores the positive aspects of measures belonging to distinct categories, which can be observed in the results of their individual optimization in the mono-objective GA. The isolated use of the IE measure results in many models with high predictive performance, but that maintain most of the features in the data sets. In many cases there is no dimensional reduction at all with the isolated use of this measure. On the other hand, the GA that uses the AC fitness function made aggressive reductions in the number of features and generated models that also stand out for their predictive power, although there are cases of significant losses in terms of accuracy.

IE + LS setting explores in a smaller scale the aggressiveness of the LS measure in reducing the percentage of features, which is similar to that obtained by the AC criterion. This behavior is emphasized, in general, by observing that this combination presents a greater number of models with accuracies statistically lower than those of the baseline models when compared to other multi-objective settings. A possible justification for this fact is that both measures belong to the distance category, what is not observed in the other combinations investigated. Thereafter, their combination do not enjoy the benefits of MO for FS regarding different categories of importance.

In addition, the LS measure is dependent on a parameter for the construction of the weight matrix of the graph that models the local structure of the data. This parameter was set in the experiments with a unique value, motivating further studies for the investigation of other values. This study may also contribute to prevent the occurrence of cases in which no feature is selected by the mono-objective GA with the LS measure, which was a specific behavior of this criterion in previous experiments which, for instance, show the problem of division by zero in Equation 3.

It is interesting to notice that the combinations between the IE measure and measures IP and RE tends to maintain all features in several cases. In fact, it was found experimentally that the measures RE and IP in general are relatively more conservative than the criteria AC and LS regarding PR, which may have contributed to the fact that these measures reached satisfactory accuracy rates in some mono-objective experiments. For monotonic criteria as IP, conservatism can be explained because a larger number of selected features may allow to define more logical hypothesis [1]. In general, it appears that the joint optimization of conservative measures, such as IE, IP and RE tends to generate models that keep that characteristic. Specifically in MOGA IE + IP and IE + RE, the strong conservatism of the IE measure prevailed over the mild conservatism of IP and RE measures.

An analysis of the results obtained by CFS and CSE shows that these algorithms are competitive with the GA investigated. The MOGA optimizing IE + AC is that which most closely approximates the results of these algorithms considering predictive performance. The number of models that are highlighted with a circle or statistically better than c_a after the application of these techniques is higher than that observed for all mono-objective and multi-objective GA, while the average PR for each one of them in the nine data sets is higher than 50%. The CSE algorithm specifically exhibits the disadvantage of presenting a larger number of models with results statistically lower than the baseline when compared to IE + AC MOGA and CFS. Importantly, both FS algorithms, like all MOGA, present in most experiments a larger number of models with no statistical difference when compared to the baseline.

We also noticed that the use of J48 provided the occurrence of a larger total number of models with statistical superiority to c_a than the SVM. It appears that the accuracy rate of models generated from projections of features is influenced by the classification technique used afterwards. This influence may have led to the fact that a FS algorithm underperforming c_a for a particular classification technique can be superior for other technique. Future work shall investigate the predictive behavior of other classification techniques with and without FS. Some initial experiments have been performed and have confirmed those observations.

In future we also plan to combine in a MOGA the IE and the importance measure of CFS (original or altered as in [26,30]). Herewith, it would be possible to perform FS considering both distance and dependency, as in IE + AC, using measures that have been investigated in recent studies [11,14,35,36,42]. In fact, the AC dependence measure explores only relevance, selecting features most correlated with the data labels. Since the measure used in CFS also considers the correlation between features, it also addresses the non-redundancy aspect.

Therefore, the method used in this work supports the implementation of different measures of importance of features, including those from algorithms CFS and CSE. This flexibility already enabled the investigation of labeled data with numerical feature values using different combinations of six criteria, taken in pairs or triplets [35,36]. Besides, some importance measures are also flexible. Measures such as IE can be used for FS in data sets with categorical features by using other distance metrics [39], while the LS and RE criteria are applicable to unlabeled data. Another advantage of the MOGA is its ability to return multiple solutions (various subsets of features). Only one of them was selected with the CP technique in our work, but others could be selected using different techniques or they could be even combined.

All FS algorithms investigated in this work belong to the filter approach. It was possible to build many models with similar accuracy to those of classifiers that use all features, while using lower numbers of features. For J48, for example, this could lead to the obtainment of decision trees with fewer nodes and can result in more understandable decision rules. These advantages are achieved with a computational cost potentially lower than would be obtained with algorithms that employ wrapper measures [25].

5 Conclusion

This work presented an evaluation of a MOGA for FS in labeled data sets, including different multi-objective configurations of features' importance measures. Their results were also compared to those of GA that optimize each importance measure individually and two popular FS techniques available in a free tool. In the experimental results and discussions we observed a prominence of the combination of IE and AC measures, coming from categories based on distance and dependency, respectively, and of the two filter algorithms from literature, which allowed to obtain different models with reduced number of features and good predictive performance.

The good results of the IE + AC MOGA in this study suggest that combining measures belonging to different categories of importance is interesting for FS in labeled data. The multi-objective optimization of these measures enables the identification of relevant features both in terms of separability between examples and correlation between features and the class. It should be also worth to analyse the degree of complementarity of these measures.

As future work we aim to combine the measures IE and that of the CFS algorithm in NSGA-II and also compare the results of different classifiers when using the subsets of features. It is also interesting to investigate the application of measures such as IE and IP in data sets with categorical features and perform experiments with the LS and RE criteria in unlabeled data sets. We can still observe the influence of different parameter values for the LS importance measure and to compare the investigated MOGA to other multi-objective metaheuristics, as well as to wrapper FS approach.

Acknowledgements. To UFABC, CAPES, FAPESP and CNPq for the support received for this work and to the staff of the IDA project for the cooperation.

References

1. Arauzo-Azofra, A., Benitez, J.M., Castro, J.L.: Consistency measures for feature selection. Journal of Intelligent Information Systems 30(3), 273–292 (2008)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Banerjee, M., Mitra, S., Banka, H.: Evolutionary rough feature selection in gene expression data. IEEE Transactions on Systems Man and Cybernetics 37(4), 622–632 (2007)
4. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA — a platform and programming language independent interface for search algorithms. In: Evolutionary Multi-Criterion Optimization, pp. 494–508 (2003)
5. Bruzzone, L., Persello, C.: A novel approach to the selection of spatially invariant features for the classification of hyperspectral images with improved generalization capability. IEEE Transactions on Geoscience and Remote Sensing 47, 3180–3191 (2009)
6. Bui, L.T., Alam, S.: An Introduction to Multiobjetive Optimization. Information Science Reference (2008)
7. Charikar, M., Guruswami, V., Kumar, R., Rajagopalan, S., Sahai, A.: Combinatorial feature selection problems. In: Annual Symposium on Foundations of Computer Science, pp. 631–640 (2000)
8. Chung, F.: Spectral Graph Theory. AMS, Providence (1997)
9. Coello, C.A.C.: Evolutionary multi-objective optimization: a historical view of the field. Computational Intelligence Magazine, 28–36 (2006)
10. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines and other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
11. Danger, R., Segura-Bedmar, I., Martínez, P., Rosso, P.: A comparison of machine learning techniques for detection of drug target articles. Journal of Biomedical Informatics, 1–12 (2010)

12. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J., Schwefel, H.P. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
13. Dessí N., Pes, B.: An evolutionary method for combining different feature selection criteria in microarray data classification. Journal of Artificial Evolution and Applications, 1–10 (2009)
14. Duangsoithong, R., Windeatt, T.: Correlation-based and causal feature selection analysis for ensemble classifiers. In: Artificial Neural Networks in Pattern Recognition, pp. 25–36 (2010)
15. Dy, J.G.: Unsupervised feature selection. In: Liu, H., Motoda, H. (eds.) Computational Methods of Feature Selection, pp. 19–39. Chapman & Hall/CRC (2008)
16. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Phd thesis, University of Waikato (1999)
17. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: International Conference on Machine Learning, pp. 359–366 (2000)
18. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann, San Francisco (2006)
19. Handl, J., Kell, D.B., Knowles, J.: Multiobjective optimization in bioinformatics and computational biology. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 279–292 (2007)
20. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing Systems, pp. 507–514 (2005)
21. Jaimes, A.L., Coello, C.A., Barrientos, J.E.U.: Online objective reduction to deal with many-objective problems. In: International Conference on Evolutionary Multi-Criterion Optimization, pp. 423–437 (2009)
22. Kruskal, W., Wallis, W.A.: Use of ranks in one-criterion variance analysis. American Statistical Association 47, 583–621 (1952)
23. Lee, H.D., Monard, M.C., Wu, F.C.: A fractal dimension based filter algorithm to select features for supervised learning. In: Advances in Artificial Intelligence, pp. 278–288 (2006)
24. Liu, H., Setiono, R.: A probabilistic approach to feature selection - a filter solution. In: International Conference on Machine Learning, pp. 319–327 (1996)
25. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/CRC (2008)
26. Lutu, P.E.N., Engelbrecht, A.P.: A decision rule-based method for feature selection in predictive data mining. Expert Systems with Applications 37(1), 602–609 (2010)
27. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge (1998)
28. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 301–312 (2002)
29. Neshatian, K., Zhang, M.: Pareto front feature selection: using genetic programming to explore feature space. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 1027–1034 (2009)
30. Nguyen, H., Franke, K., Petrovic, S.: Improving effectiveness of intrusion detection by correlation feature selection. In: International Conference on Availability, Reliability and Security, pp. 17–24 (2010)
31. QuinLan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)

32. Salzberg, S.L.: On comparing classifiers: Pitfalls to avoid and a recommended approach. Data Mining and Knowledge Discovery 1, 317–328 (1997)
33. Santana, L.E.A., Silva, L., Canuto, A.M.P.: Feature selection in heterogeneous structure of ensembles: a genetic algorithm approach. In: International Joint Conference on Neural Networks, pp. 1491–1498 (2009)
34. Shon, T., Kovah, X., Moon, J.: Applying genetic algorithm for classifying anomalous tcp/ip packets. Neurocomputing 69, 2429–2433 (2006)
35. Spolaôr, N., Lorena, A.C., Lee, H.D.: Seleção de atributos por meio de algoritmos genéticos multiobjetivo (in portuguese). In: Workshop on MSc Dissertation and PhD Thesis in Artificial Intelligence, pp. 1–10 (2010)
36. Spolaôr, N., Lorena, A.C., Lee, H.D.: Use of multiobjective genetic algorithms in feature selection. In: IEEE Brazilian Symposium on Artificial Neural Network, pp. 1–6 (2010)
37. Wang, C.M., Huang, Y.F.: Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data. Expert Systems with Applications 36(3), 5900–5908 (2009)
38. Wang, L., Fu, X.: Data Mining With Computational Intelligence. Springer, Heidelberg (2005)
39. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
40. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
41. Yan, W.: Fusion in multi-criterion feature ranking. In: International Conference on Information Fusion, pp. 01–06 (2007)
42. Zaharie, D., Holban, S., Lungeanu, D., Navolan, D.: A computational intelligence approach for ranking risk factors in preterm birth. In: International Symposium on Applied Computational Intelligence and Informatics, pp. 135–140 (2007)
43. Zeleny, M.: An introduction to multiobjetive optimization. In: Cochrane, J.L., Zeleny, M. (eds.) Multiple Criteria Decision Making, pp. 262–301. University of South Carolina Press (1973)
44. Zhu, Z., Ong, Y.S., Kuo, J.L.: Feature selection using single/multi-objective memetic frameworks. In: Goh, C.K., Ong, Y.S., Tan, K.C. (eds.) Multi-Objective Memetic Algorithms, pp. 111–131. Springer, Heidelberg (2009)

A Cultural Algorithm Applied in a Bi-Objective Uncapacitated Facility Location Problem

Guillermo Cabrera, José Miguel Rubio, Daniela Díaz, Boris Fernández,
Claudio Cubillos, and Ricardo Soto

Pontificia Universidad Católica de Valparaíso, Escuela de Ingeniería Informática,
Av. Brasil 2241, Valparaíso, Chile
{guillermo.cabrera,jose.rubio.l,claudio.cubillos,ricardo.soto}@ucv.cl,
{daniela.diaz.d,boris.fernandez}@mail.ucv.cl
http://www.inf.ucv.cl

Abstract. Cultural Algorithms (CAs) are one of the metaheuristics which can be adapted in order to work in multi-objectives optimization environments. On the other hand, Bi-Objective Uncapacitated Facility Location Problem (BOUFLP) and particularly Uncapacitated Facility Location Problem (UFLP) are well know problems in literature. However, only few articles have applied evolutionary multi-objective (EMO) algorithms to these problems and articles presenting CAs applied to the BOUFLP have not been found. In this article we presents a Bi-Objective Cultural Algorithm (BOCA) which was applied to the Bi-Objective Uncapacitated Facility Location Problem (BOUFLP) and it obtain an important improvement in comparison with other well-know EMO algorithms such as PAES and NSGA-II. The considered criteria were cost minimization and coverage maximization. The different solutions obtained with the CA were compared using an hypervolume S metric.

Keywords: Bi-Objective Cultural Algorithm, Bi-Objective Uncapacitated Facility Location Problem, Evolutionary Multi-Objective Optimization, S metric.

1 Introduction

In order to solve several complex optimization problems, the evolutionary algorithms have become an efficient and effective alternative for researchers principally because this kind of techniques is capable to found good solutions for most of these problems in acceptable computational times. By above, it is reasonable to think that if these algorithms are used we can reach goods solution in a multi-objective environment, in a similar way as in mono-objective optimization problems. However, the difficulties that these algorithms present (particularly genetic algorithms) in a mono-objective environment could also happen in multi-objective environments. Specifically, they easily fall into premature convergence with low evolution efficiency because implicit information embodied in the evolution process and domain knowledge corresponding to optimization problems is

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 477–491, 2011.

not fully used [3]. In order to effectively make use of implicit evolution information, in [4] the author proposed CAs which were inspired from human culture evolution process.

CAs haves a dual evolution structure which consists in both spaces: population and belief space. The population space works as any other evolutionary algorithm. However, in the belief space, implicit knowledge is extracted from better individuals in the population and stored in a different way. Then, they are used to guide the evolution process in the population space so as to induce population escaping from the local optimal solutions. It has been proved that cultural algorithms can effectively improve the evolution performance. And the algorithms also provide a universal model for extraction and utilization of the evolution information [3].

In the last 20 years, several authors have centered their efforts in the development of several EMO algorithms in order to solve a specific group of problems which are called Multi-Objective or, generalizing, Multi-Criteria[1]. Among these works we can cite [5] where the author uses a genetic algorithm in order to solve the multi-objective dynamic optimization for automatic parking system, [6] where the authors proposes an improvement to the well-know NSGA algorithm (and that they called NSGA-II) based on an elitist approach, and [9] where the author presents an EMO algorithm applied for a specific variation of the well-studied CVRP, where the author includes in the EMO algorithm an explicit collective memory method, namely the extended virtual loser (EVL). In [14] a complete and extensive literature review related with EMO can be found. Despite the above, we do not found in literature articles where the CAs were applied to BOUFLP, In fact, some recent published books [26] and [27] only mention Coello and Landa investigations [16] as an example of CA application to solve MOPs.

In this article we present a Bi-Objective Cultural Algorithm (BOCA) which was applied to the Bi-Objective Uncapacitated Facility Location Problem (BOUFLP) and it obtains an important improvement in comparison with other well-know EMO algorithms as PAES and NSGA-II. The criterion considered in this work included both, cost (in order to minimize it) and coverage (in order to maximize it). The different solutions obtained with the CA were compared using an S metric proposed in [1] and used in several works in the literature.

1.1 Paper Organization

In section 2 we show an overview about multi-objective optimization, emphasizing in EMO algorithms, in sub-section 2.2 we presents a BOUFLP model and some distinctive characteristics of them, and finally we describe the BOCA algorithm and its principal characteristics. In section 3 we describe in detail the properties of multi-objective optimization, particularly the non-dominated sets. In this section, the explanation of how to construct the non-dominated sets and how to compare different non-dominated sets is depicted. In section 4 we present

[1] In this paper both terms are used indistinctly.

the BOCA applied to well-know instances of literature, its results and a discussion about them. Finally, in section 5 we present our conclusions and future work.

2 Overview

In this section we show an overview of topics related to this paper. In sub-section 2.1, the multi-objective optimization is reviewed, emphasizing in EMO algorithms and its state-of-art. Particularly, we focused in the development of EMO algorithms for Multi-Objective Combinatorial Optimization problems. In sub-section 2.2 we review the BOUFLP and present its model formulation based in a cost-coverage approach. Finally, in sub-section 2.3 we briefly present the traditional CAs and we show a complete review regarding about multi-criterion CAs in literature and the details of our CA. implementation.

2.1 (Evolutionary) Multi-Objective Optimization

In this section we briefly introduce the main principles of MOP and, particularly, MOCO problems. The Following definitions were extracted from [22]. Given a set of alternatives, a feasible alternative x is called *dominated* if there is another feasible alternative in the set, say alternative x', such that:

- x' is equally or more preferred than with respect to all criteria, and
- x' is more preferred than for at least one criterion

If the above holds, the alternative x' is called dominating. A pair of alternatives x and x', where x is dominated and x' in dominating, is said to be in *pareto dominance relation* and is denoted by $x' \preceq x$. In a set of more than two alternatives, one alternative can be dominating and dominated at the same time. Given a set of feasible alternatives, one which is not dominated by any other alternative of this set is called efficient. In other words, an alternative is efficient if there is no other alternative in the set:

- equally or more preferred with respect to all criteria, and
- more preferred for at least one criterion

Alternatives which are not efficient are called *nonefficient*. The *pareto optimal* set (P^*) is composed by feasible solutions which are not dominated by any other solution. Therefore, $P^* = \{x \in \Omega : \textit{there is no } x' \in \Omega, x' \preceq x\}$.

The Efficient (or Pareto) Frontier (PF^*) is the image of the pareto optimal set in the objective space, this is, $PF^* = \{f(x) = (f_1(x)...f_k(x)) : x \in P^*\}$.

As said above, many authors have worked in order to solve different MOCO problems. Particularly, EMO is an important research area for this goal. An evolutionary algorithm is a stochastic search procedure inspired by the evolution process in nature. In this process, individuals evolve and the fitter ones have a better chance of reproduction and survival. The reproduction mechanisms favor the characteristics of the stronger parents and hopefully produce better children guaranteeing the presence of those characteristics in future generations [2]. A complete review about different EMO algorithms is presented in [19], [26] and, more recently, in [27].

2.2 Bi-Objective Uncapacitated Facility Location Problem (BOUFLP)

Facility Location Problems (FLP) is one of the most important problems for companies with the aim of distributing products to their customers. The problem consists of selecting sites to install plants, warehouses, and distribution centers, assigning customers to serving facilities, and interconnecting facilities by flow assignment decisions. In [10], [11] and [12] detailed FLP reviews and analysis are presented.

This paper considers a two-level supply chain, where a single plant serves a set of warehouses, which serve a set of end customers or retailers. Figure 1 shows this configuration.

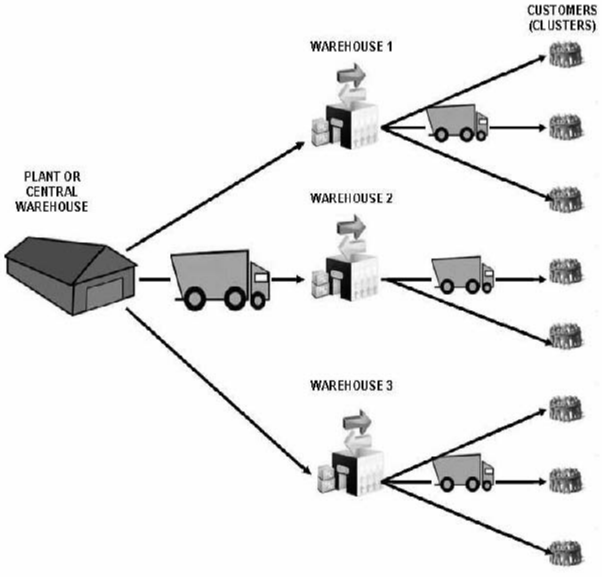


Fig. 1. A two-level supply chain network configuration

This model focused in two main goals:

- Minimize the total cost associated with the facility installation and customer allocation or
- Maximize the customer's rate coverage.

Respect to the minimization of location and allocation costs, uncapacitated FLP (UFLP) is one of most studied models in literature. Specifically, in the UFLP model, the main characteristic is defined for the unlimited capacity of the distribution centers which permits them to serve an unlimited number of customers, independently of its demand. On the other hand, respect to the coverage, the Maximal Covering Location Problem (MCLP) is our reference model. Both models are well-described and formalized by Daskin in [10]. In this context, a classification developed in [8], shows how to relate a multi-criterion optimization problem with a specific model as a BOUFLP. Figure 2 shows such classification.

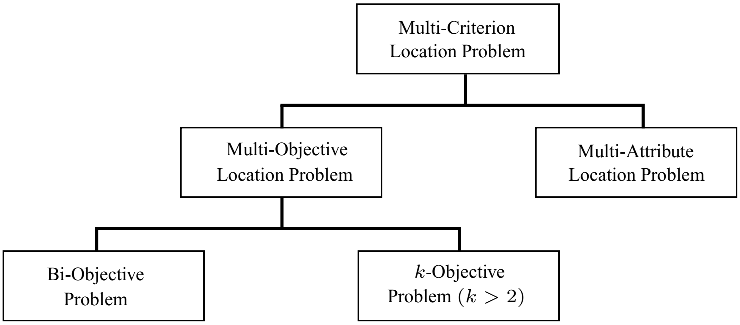

Multi-Criterion Location Problem
Multi-Objective Location Problem
Multi-Attribute Location Problem
Bi-Objective Problem
k-Objective Problem ($k > 2$)

Fig. 2. A classification of Multi-criterion Location Problem presented in [8]

In this paper, as mentioned above, we solved the BOUFLP. The problem has been modeled with mini-sum and maxi-sum objectives (cost and coverage).

The following model formulation is based on [13]. Let $I = \{1, ..., m\}$ be the set of potential facilities and $J = \{1, ..., n\}$ the set of customers. Let f_i be the fixed cost of operating facility i and d_j the demand of customer j. Let c_{ij} be the cost of assigning the customer j to facility i and h_{ij} the distance between facility i and customer j. Let D_{MAX} be the maximal covering distance, that is, customers within this distance to an open facility are considered well served. Let $Q_j \triangleq \{i \in I : h_{ij} \leq D_{MAX}\}$ be the set of facilities that could attend customer j within the maximal covering distance D_{MAX}. Let y_i be 1 (one) if facility i is open and 0 (zero), otherwise. Let x_{ij} be 1 (one) if the whole demand of customer j is attended by facility j and 0 (zero), otherwise.

$$min\ z_1 = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i \tag{1}$$

$$max\ z_2 = \sum_{j \in J} d_j \sum_{i \in Q_j} x_{ij} \tag{2}$$

subject to,

$$\sum_{i \in I} x_{ij} = 1,\ \forall j \in J \tag{3}$$

$$x_{ij} \leq y_i, \ \forall j \in J, \forall i \in I \tag{4}$$

$$x_{i,j} \in \{0,1\} \ \forall j \in J, \forall i \in I \tag{5}$$

$$y_i \in \{0,1\} \ \forall i \in I \tag{6}$$

The objective function (1) represents total operating cost; where the first term represents the allocation cost (cost of attending demand of the customer by the open facilities), and the second term, the location cost (sum of the fixed costs incurred by the open facilities). The objective function (2) measures coverage as the sum of the demand of customers attended by open facilities within the maximal covering distance. Constraints (3) and (5), guarantees that each customer is attended by only one facility. Constraint (4) forces the customer to be assigned to an open facility. Finally, (5) and (6) define decision variables as binary.

2.3 Bi-Objective Cultural Algorithms (BOCA)

The experience and beliefs accepted by a community in a social system are the main motivations for the creation of Cultural Algorithms (CAs). They were developed by Robert Reynolds, in order to model the evolution of cultural systems based on the principles of human social evolution from the literature of social sciences, who believe that the evolution can be seen as an optimization process [4]. The CAs are identified to guide the evolution of the population based on the knowledge. This applies to the knowledge provided to future generations, allowing them to accelerate the convergence of the algorithm to obtain good solutions [21]. Besides the domain knowledge is modeled separately from the population, because there is certain independence between both, which allow to work and to model separately each one of them, in order to enhance the overall algorithm and thus improve the search for best solutions. Figure 3 shows this interaction.

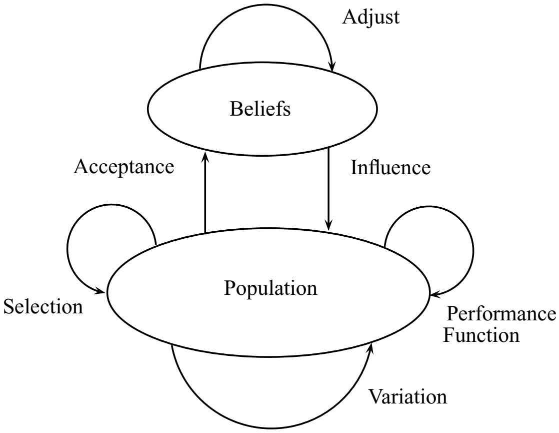


Fig. 3. Spaces of a Cultural Algorithm

CAs are mainly characterized by presenting two inheritance systems: at the population level and knowledge. This key feature is designed to increase the learning rates and convergence of the algorithm, and thus to do a more responsive system for a number of problems [23]. This feature allows to identify two significant levels of knowledge: a micro-evolutionary level (represented by the area of population) and macro-evolutionary level (represented by the space of beliefs) [24]. CAs have the following components: population space (set of individuals who have independent features) [24]; belief space (stored knowledge individuals have acquired in previous generations) [23]; computer protocol, which connecting the two spaces and defining the rules on the type of information to be exchanged between the spaces, by using the acceptance and influence function; and finally, knowledge sources which are described in terms of their ability to coordinate the distribution of individuals on the nature of an instance of a problem [24]. These knowledge sources can be of the following types: *circumstantial, normative, domain, topographic, historical.*

Algorithm 1. General Structure of BOCA

```
procedure General Structure of BOCA
    $t = 0$
    initialize Population $P(t)$
    initialize belief Space $B(t)$
    evaluate individual fitness of $P(t)$
    while not end condition do
        $t = t + 1$
        $parents = select\ parents\ from\ P(t-1)\ and\ Influence\ from\ B(t)$
        $child = crossover(parents, Influence(B(t))$
        $Evaluate(child)$
        $P(t) = child$
        $Update(B(t))$
        $Accept(P(t))$
    end while
    while not end condition do
        $i = i + 1$
        $Baux(i) = Mutate(B(t))$
        $Evaluate(Baux(t))$
        $Update(B(t))$
        $Accept(Baux(t))$
    end while
end procedure
```

BOCA Implementation. In this article, a binary representation was chosen, with an *m-length* vector (with m a number of possible facilities) where a position $i = 1$ implies that the facility i has been selected. Besides, we used a $m \times n$ matrix (with m a number of possible facilities and n a number of customers) where a position $(i, j) = 1$ implies that the customer j is served by facility i.

In order to compare the two different solutions (individuals) it is necessary a evaluation criterion. As mentioned in previous sections, this criterion is difficult to define due to the multi-objective nature of the environment in which we are working. Given the dominance concept explained in Section 2.1, many individuals are not comparable only using the objective functions of the problem. To solve this problem, we propose a fitness function (formula (7)) which will determine which individual is "better" than another.

$$fitness = \left[\frac{gap_1 \times 100}{ub_1 - lb_1} + \frac{gap_2 \times 100}{ub_2 - lb_2}\right] \tag{7}$$

Where gap_1 is the difference between the values of function z_1 and lb_1, which is the lower bound of function z_1. The gap_2, meanwhile, is the difference between the value of function z_2 and lb_2, which is the lower bound of function z_1. Finally, ub_1 and ub_2 are the upper bounds of z_1 and z_2 function respectively. Figure 4 shows the relation between these values.

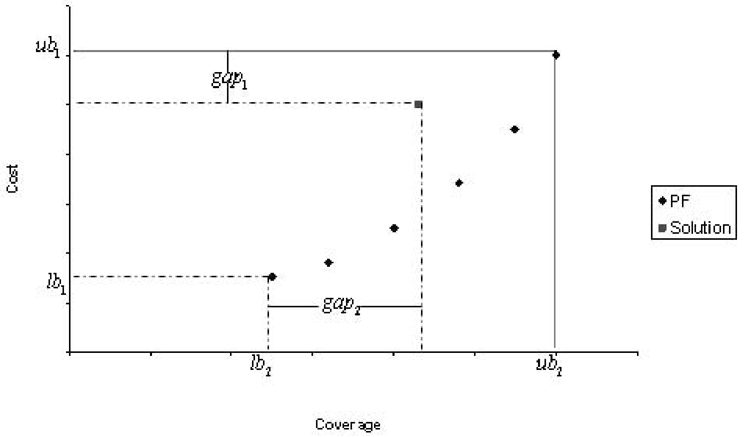


Fig. 4. Main coordinates in order to obtain a fitness value for an individual

In order to initialize the population, we used a semi-random function. In first phase, this function defines in a stochastic way the set of facilities which will be open (selected facilities). Then, we allocate the customer into a selected facility minimizing the cost function and avoiding minimizing a coverage function. This strategy for assigning the initial population provided better results than using initial populations generated with a completely random function, and the computational-time cost of this semi-random function was marginal.

The most distinctive feature of CAs is the integration of knowledge, which through an influence function affects future generations. In this paper, the influence functions that were used are those based on circumstantial and historical knowledge. This is used for best individuals found so far adapted (circumstantial) and a list of individuals who possess the highest level of adaptation (lower fitness value) compared to the rest of the population (historical). In order to obtain a next generation, two parents are used in a recombination process. To avoid local optimal values, we do not overuse the culture. Hence, a parent is selected from the population in order to obtain diversity and the other parent is selected from the knowledge to influence the next generation. Historical knowledge keeps a list of all individuals found close to the Pareto Frontier and that will be used as parents of future generations. Circumstantial Knowledge provides information from the two best individuals found so far. An individual will give information on the best value found for z_1 and z_2 individual to another.

3 Non-dominated Sets: Metrics

In this section we face one of most important problems in MOCO: How can we compare two non-Dominated Sets? Several proposals from literature and its main characteristics are described, in order to choose one of them and applied over the non-dominated sets obtained for the BOCA algorithm.

3.1 Metrics for Comparing NDSs

As said above, one of most important problems in MOCO is how to compare two NDSs. Numerous quality assessment metrics have been developed by researchers to compare the performance of different MOEA. These metrics show different properties and address various aspects of solution set quality [17].

In this sense [17] describes several *Excellence Relations*. These relations establish strict partial orders in the set of all NDSs with respect to different aspects of quality. Previously, in [18] and [19] the authors consider several Outperformance Relations to address the closeness of NDSs to the Pareto Frontier (PF). Besides the above, also is necessary to measure in a quantitative way the approximation to PF. In this sense, we need to identify desirable aspects of NDSs. In [20] the authors define those desirable aspects:

- The distance of the resulting non-dominated set to the Pareto-optimal frontier should be minimized.
- A good (in most cases uniform) distribution of the solutions found is desirable. The assessment of this criterion might be based on a certain distance metric.
- The extension of the obtained non-dominated frontier should be maximized, i.e., for each objective, a wide range of values should be covered by the non-dominated solutions.

The main problem of these, is that the three criteria should be combined in different ways to establish the performance of an EMO algorithm in quantitative way. However, such combinations are simply linear combinations of weights. Ironically, the problem of metrics is also multi-objective nature. Over the last two decades, several authors have described techniques for measuring one or more of these criteria and also, they have proposed new criteria that give new insights for the evaluation of all the solutions found.

In this article, we choose the S metric. In [19] the formal definition for the S metric can be found. This metric calculates the hyper-volume of the multidimensional region[2] [1] and allows the integration of aspects that are individually measured by other metrics. In this article, the total area is bounded by the points where the maximum coverage (all centers are open) and minimum cost (the cheaper location are open) are reached. It is easy to note that, given the mentioned area, the ideal point (maximum coverage at minimum cost) will have a value for the metric equivalent to S = 1 (equivalent to 100% of the area). An advantage of the S metric is that each MOEA can be assessed independently of the other MOEAs. However, the S values of two sets A, B cannot be used to derive whether either set entirely dominates the other. Fortunately, this disadvantage does not affect the metric used in this article.

4 Computational Experiments

In this section we present the instances of literature and the results obtained for them by our CA implementation. Sub-section 4.1 shows the main characteristics of the used instances and sub-section 4.2 presents the main results and a brief discussion. All results are shown in appendix A.

4.1 Instances Presentation

The instances that were used in this research are the ones generated by [2]. The descriptions of instances were extracted from that article. These instances correspond to random instances using a problem generator that follows the methodology of UflLib [25].

Instances were generated of three different sizes ranging from 10 facilities and 25 customers, to 50 facilities and 150 customers (problem sizes $m \times n$: 10×25, 30×75 and 50×150). Instances have two different ways of choosing the facility location (type A and B). Instances of type A choose uniformly distributed facility locations within a square of 190 distance units of width; while instances of type B choose facility locations from customer locations, thus facility locations correspond to customer locations.

We generated problem instances with 6 different types of fixed-cost structures (C1 to C6). Instances labeled C1, C2, and C3, have the same fixed cost for all facilities; namely, 400, 700, and 1000, respectively. Instances labeled C4, C5, and C6, have uniformly distributed fixed costs for each facility; namely, U(100, 400), U(400, 700), and U(700, 1000), respectively (where U(a, b) stands for a uniformly distributed random variable between a and b).

After combining all the possibilities of size, facility location and cost structure, we end up with 36 problem instances. Each problem instance was given a name

[2] In this paper the metric calculates the area of the two-dimensional region.

following the (A or B) m - n (C1 to C6). For example, instance A50-150C2 has its 50 facilities uniformly distributed within the square of width 190 distance units; has 150 customers; and facility fixed-cost structure of type C2.

On the other side, the cultural algorithm proposed in this research has the following parameters to be set: population size L, number of generations T, probability of mutation in the population P_{mp} and probability of mutation in the belief space P_{mb} . To make the parameter setting, we used four test problems and several values for each parameter, which are showed in Table 1:

Table 1. Test values for parameters setting

Parameter	**Test Values**
L	30 - 60 - 90 - 120 - 150
T	50 - 100 - 150 - 200 -250 - 300
P_{mp}	0.1 - 0.15 - 0.2 - 0.25 - 0.3
P_{mb}	0.02 - 0.03 - 0.04 - 0.05

The algorithm was executed 10 times for each combination of parameters and calculates the average for each of them. Finally the chosen parameters correspond to those that, together, yielded the best results for the four instances. Table 2 shows the values used for the experiments.

Table 2. Parameters setting

Parameter	**Set Value**
L	150
T	100
P_{mp}	0.15
P_{mb}	0.04

4.2 Results and Discussion

In order to resume our results, we only present the ones obtained for six instances in which CA obtained an important improvement regarding the NSGA-II and PAES algorithm.

Figure 5 shows the behavior of our BOCA implementation and its results before exploitation phase and after exploitation phase. As can we see, this phase allows to obtain a robust approximation with more non-dominated solutions (allows a compact approximation) and more dominated area. By above, we can say that our BOCA implementation uses the belief space in a good way, and that the historical knowledge used is the most important, responsible of this improvements regarding the results before exploitation phase.

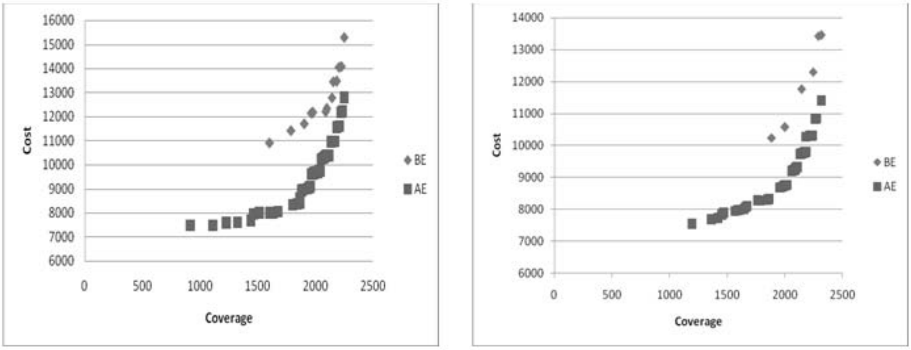


Fig. 5. Approximation to Pareto Frontier: Before Exploitation (BE) phase and After Exploitation phase for (a) A30-75C2 and (b) B30-75C2 instances

Tables 3 and 4 presents a comparison between our BOCA implementation and two well-know MOEAs: NSGA-II and PAES. The results of these implementations are obtained from [2], where these algorithms were applied to BOUFLP, as said above. Table 1 shows a BOCA v/s PAES comparison. The main conclusion of this comparison is that our BOCA implementation obtained an important improvement when compared with the PAES algorithm. Furthermore, a slight improvement is obtained regarding computational times. The average of the improvement for the S-area for all instances was a 7.9%, and for the computational times was a 1.97%.

Table 3. Comparison between BOCA and PAES algorithms

	BOCA			PAES			
Instance	S'_{BOCA}	Time(sec).	$\|PF^*\|$	S'_{PAES}	Time(sec).	$\|PF^*\|$	**Difference**(%)
A10-25C1	0,7057	252	13	0,7057	372	13	0
A30-75C6	0,7245	740	37	0,5300	511	12	36,70
A50-150C3	0,7883	1165	53	0,6900	1269	26	14,25
B10-25C6	0,6517	285	18	0,6517	394	17	0
B30-75C6	0,8775	800	42	0,7602	585	32	15,43
B50-150C2	0,7634	1221	53	0,4970	1241	13	53,60

Table 4 shows a BOCA v/s NSGA-II comparison. The main conclusion of this comparison is that our BOCA implementation obtained an important improvement regarding to the NSGA-II algorithm if we compare a computational times. However, the improvement obtained when considering the S-area is marginal. The average of the improvement for the S-area considering all instances was a 0.4%, and for the computational times was a 60.74%. This improvement (computational time) can be attributed to the exploitation phase, which is more "aggressive" than GA, allowing a more quick convergence. The best improvements in the S-area obtained by our BOCA implementation are in the following two instances: A30-75C6 and B30-75C6, with 8.52% and 8.49% respectively. As can we see, both instances have similar characteristics. This can explain the good BOCA performance.

Table 4. Comparison between BOCA and NSGA-II algorithms

	BOCA			NSGA-II			
Instance	S'_{BOCA}	Time(sec).	$\lvert PF^*\rvert$	$S'_{NSGA-II}$	Time(sec).	$\lvert PF^*\rvert$	**Difference**(%)
A10-25C1	0,7057	252	13	0,7057	1344	13	0
A30-75C6	0,7245	740	37	0,6676	1474	36	8,52
A50-150C3	0,7883	1165	53	0,7665	2298	34	2,84
B10-25C6	0,6517	285	18	0,6517	1261	17	0
B30-75C6	0,8775	800	42	0,8088	1460	47	8,49
B50-150C2	0,7634	1221	53	0,7515	2289	41	1,58

5 Conclusions and Future Work

In this work, we have confirmed the efficiency of CAs w.r.t. other evolutionary techniques such as genetic algorithms. This better performance of CAs has been widely reported in mono-objective problems, however the research in a multi-objective context has not achieved that maturity level yet.

The main goal of this article was to study the behavior and performance of an implementation of a CA in a multi-objective environment. In this sense, our implementation has achieved the expected results. Thus, confirming that the CAs are a real alternative and a line of development that has not been sufficiently developed.

The results obtained by our implementation, show significant improvements over the two dimensions measured in this paper: dominated area (S-metric) and computational time. Our implementation was contrasted with two well known algorithms in the literature: PAES and NSGA-II. In the first case (PAES), the most significant improvements occurred in the dimension of dominated area, while the second (NSGA-II) the most important improvements concerned the computational time. As future work, we envision two lines of work that are relevant and presented below:

- Regarding the *type of knowledge* used. In this investigation two types of knowledge have been inserted in the space of belief: the circumstantial and historical. This opens a line of work for experimenting with other kinds of knowledge to further improve the performance of the algorithm.
- Regarding the metrics. It seems interesting to evaluate the performance of cultural algorithm using other metrics to measure the level of consistency and quality of the approximations obtained.

References

1. Knowles, J., Corne, D.: On Metrics for Comparing Nondominated Sets. In: Proceedings of the Congress on Evolutionary Computation, vol. 1, pp. 711–716 (2002)
2. Villegas, J.G., Palacios, F., Medaglia, A.L.: Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. Annals of Operations Research 147(1), 109–141 (2006)
3. Guo, Y.-n., Cheng, J., Cao, Y.-y., Lin, Y.: A novel multi-population cultural algorithm adopting knowledge migration. Soft Computing - A Fusion of Foundations, Methodologies and Applications. Springer, Heidelberg (2010) (in press)
4. Reynolds, R.G.: An Introduction to Cultural Algorithms. In: Proceedings of the 3rd Annual Conference on Evolutionary Programming, pp. 131–139. World Scienfific Publishing, Singapore (1994)
5. Maravall, D., de Lope, J.: Multi-objective dynamic optimization with genetic algorithms for automatic parking. In: Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 11(3), pp. 249–257 (2007)
6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimisation: NSGA-II. In: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, pp. 849–858 (2000)
7. Carrano, E.G., Takahashi, R.H.C., Fonseca, C.M., Neto, O.M.: Bi-objective Combined Facility Location and Network Design. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 486–500. Springer, Heidelberg (2007)
8. Farahania, R.Z., SteadieSeifib, M., Asgaria, N.: Multiple criteria facility location problems, A survey. Applied Mathematical Modelling 34(7), 1689–1709 (2010)
9. Borgulya, I.: An algorithm for the capacitated vehicle routing problem with route balancing. Central European Journal of Operations Research 16(4), 331–343 (2008)
10. Daskin, M.: Network and discrete location, Models: algorithms, and applications. Wiley Interscience, New York
11. Simchi-Levi, D., Chen, X., Bramel, J.: The logic of logistic. Springer, New York
12. Drezner, Z., Hamacher, H.W. (eds.): Facility Location: Applications and Theory. Springer, New York (2002)
13. Revelle, C.S., Laporte, G.: The Plant Location Problem: New Models and Research Prospects. Operations Research 44(6), 864–874 (1996)
14. Bhattacharya, R., Bandyopadhyay, S.: Solving conflicting bi-objective facility location problem by NSGA II evolutionary algorithm. The International Journal of Advanced Manufacturing Technology (2010) (in press)
15. Gu, W., Wu, Y.: Application of Multi-objective Cultural Algorithm in Water Resources Optimization. Power and Energy Engineering Conference (APPEEC), 1–4 (2010)
16. Coello, C.A.C., Becerra, R.L.: Evolutionary Multiobjective Optimization using a Cultural Algorithm. In: IEEE Swarm Intelligence Symposium Proceedings, pp. 6–13 (2003)
17. Farhang-Mehr, A., Azarm, S.: Minimal sets of quality metrics. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 405–417. Springer, Heidelberg (2003)
18. Hansen, M.P., Jaszkiewicz, A.: Evaluating the Quality of Approximations to the Non-dominatedSet. Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark (1998)

19. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization, Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (November 1999)
20. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms Empirical Results. Evolutionary Computation 8, 173–195 (2000)
21. Reynolds, R.G.: Cultural algorithms: Theory and applications. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 367–377. McGraw-Hill, New York (1999)
22. Kaliszewski, I.: Soft Computing for Complex Multiple Criteria Decision Making. Springer, Berlin (2006)
23. Becerra, R.L., Coello, C.A.C.: A Cultural Algorithm with Differential Evolution to Solve Constrained Optimization Problems. In: IBERAMIA, pp. 881–890 (2004)
24. Soza, C., Becerra, R.L., Riff, M.-C., Coello, C.A.C.: A Cultural Algorithm with Operator Parameters Control for Solving Timetabling Problems. In: IFSA, pp. 810–819
25. Hoefer, M.: UflLib, Benchmark Instances for the Uncapacitated Facility Location Problem, http://www.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/
26. Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
27. Coello, C.A.C., Dhaenens, C., Jourdan, L. (eds.): Advances in Multi-Objective Nature Inspired Computing. Springer, Heidelberg (2010)

A Bi-objective Iterated Local Search Heuristic with Path-Relinking for the p-Median Problem

José E.C. Arroyo, André G. Santos, Paula M. dos Santos,
and Wellington G Ribeiro

Departamento de Informática, Universidade Federal de Viçosa
Campus Universitário da UFV, 36570-00 Centro Viçosa, MG, Brasil
{jarroyo,andre}@dpi.ufv.br, paula-marianna@hotmail.com,
eltinguitarra@hotmail.com

Abstract. In this paper, we examine the p-median problem from a bi-objective point of view. Since this is a NP-Hard problem, an efficient algorithm based on the Iterated Local Search heuristic (ILS) is proposed to determine non-dominated solutions (an approximation of the Pareto-optimal solutions). ILS is a simple and powerful stochastic method that has shown very good results for a variety of single-objective combinatorial problems. In each component of the ILS, we use the concept of Pareto dominance. An intensification component based on the Path-Relinking is used to improve the quality of the found non-dominated solutions. To test the performance of the proposed algorithm, we develop a Mathematical Programming Algorithm, called ε-Constraint, that finds a subset of Pareto-optimal solutions by solving iteratively the mathematical model of the problem with additional constraints. The results show that the proposed approach is able to generate good approximations to the non-dominated frontier of the bi-objective problem efficiently.

Keywords: Heuristics, Local search, Path Relinking, Multi-objective Combinatorial Optimization, p-Medians.

1 Introduction

Given *n* customers and *m* potential facilities, the single-objective *p*-median problem (SO-*p*-MP) is the most well-known facility location problem. It consists in finding a subset with *p* facilities (medians) such that the total sum of distances between each customer and its nearest facility is minimized. Facility location-allocation problems have several applications in telecommunications, industrial transportation and distribution, clustering, etc.

The SO-*p*-MP is NP-hard [7]. Exact and heuristic methods have been proposed to solve this problem [11][12].

In this paper, we address the *p*-MP from a bi-objective point of view. Consider *m* location points (for installing or opening facilities), *n* customers, a $n \times m$ matrix with the distances traveled d_{ij} for satisfying the demand of the customer located at *i* from the facility located at *j* and the fixed cost c_j of operating local *j* (the cost of installing or opening a facility at the local *j*). The bi-objective *p*-MP (BO-*p*-MP) consists in finding a subset with *p* facilities (medians) in order to minimize, simultaneously two

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 492–504, 2011.

objectives, f_1: the sum of the distances from each customer to its nearest facility and f_2: the total cost of opening the p facilities.

A solution $\boldsymbol{x}$ to a bi-objective problem can be described in terms of decision vector in decision space X. For the p-MP, a solution $\boldsymbol{x}$ is interpreted as a set of p open facilities (or medians). The vector-valued function $\boldsymbol{f} = (f_1, f_2)$: $X \rightarrow Z$ evaluates the quality of a specific solution by assigning it an objective vector in the objective space Z ($Z \subseteq \mathbf{R}^2$). Following the well-known concept of Pareto dominance, we can say that $\boldsymbol{x}^1$ *dominates* $\boldsymbol{x}^2$ if no component of $\boldsymbol{f}(\boldsymbol{x}^1)$ is larger than the corresponding component of $\boldsymbol{f}(\boldsymbol{x}^2)$ and at least one component is smaller, that is, $f_1(\boldsymbol{x}^1) \leq f_1(\boldsymbol{x}^2)$, $f_2(\boldsymbol{x}^1) \leq f_2(\boldsymbol{x}^2)$ and ($f_1(\boldsymbol{x}^1) < f_1(\boldsymbol{x}^2)$ or $f_2(\boldsymbol{x}^1) < f_2(\boldsymbol{x}^2)$) [20].

Here, optimal solutions, i.e., solutions not dominated by any other solution may be mapped to different objective vectors. In other words, there may exist several optimal objective vectors representing different tradeoffs between the objectives. The set of optimal solutions in the decision space is in general denoted as *Pareto-optimal set.*

A multi-objective optimization problem consists in determining the set of Pareto-optimal solutions. Knowledge about this set helps the decision maker in choosing the best compromise solution.

The most used methods for solving multi-objective combinatorial optimization problems are metaheuristics [3][6]. Metaheuristic methods were originally conceived for single-objective optimization and the success achieved in their application to a very large number of problems has stimulated researchers to extend them to multi-objective combinatorial optimization problems. Surveys of multi-objective metaheuristics research reveal that in most articles from the literature use Genetic Algorithms (GAs) as the primary metaheuristic, followed by Simulated Annealing and Tabu Search [6]. Applications of others metaheuristics, such as GRASP, Iterated Local Search (ILS), VNS and Ant-Colony are scarce. Recently, these metaheuristics were applied to solve some multi-objective combinatorial optimization problems: GRASP [1], ILS [4], VNS [8], Ant-Colony [15].

The literature on multi-objective facility location-allocation problems is too scarce. Recently, GAs were used to solve some of these problems [10][14][19].

The purpose of this paper is to present a simple and effective bi-objective method based on the ILS metaheuristic to determine a good approximation of the Pareto-optimal solutions. In the proposed heuristic we use an efficient intensification procedure based on the Path Relinking technique that explores a trajectory that connects two non-dominated solutions.

2 A Bi-objective ILS Heuristic with Path Relinking

Iterated Local Search (ILS) [9] is a simple and generally applicable heuristic that iteratively applies local search to modifications of a current solution. Four basic procedures are needed to derive an ILS algorithm: a procedure "Generate-Initial-Solution", which returns an initial solution $\boldsymbol{x}$, a procedure "Perturbation" that perturbs the current solution $\boldsymbol{x}$ leading to some intermediate solutions $\boldsymbol{x}^1$, a procedure "Local-Search" that takes $\boldsymbol{x}^1$ to a local optimum $\boldsymbol{x}^2$, and an acceptance criterion that decides from which solution the next perturbation step is applied. Usually, the solution $\boldsymbol{x}^2$ returned by the local search is accepted if it is better than the current solution $\boldsymbol{x}$. ILS is a powerful heuristic that has shown very good results for a variety of single-objective combinatorial problems [9].

In this paper we present an adaptation of the ILS heuristic to solve the BO-*p*-MP. The proposed algorithm was inspired by the algorithm proposed in [16]. In the proposed ILS heuristic, we use local search and intensification with Path Relinking. The Path Relinking technique connects a non-dominated solution with an ILS local

```
BOILS+PR ($N_s$, $k$, $Max\_Iter$)
$D = \emptyset$; //set of non-dominated solutions to be  returned by BOILS+PR
Generate-Initial-Solutions:
For $i = 1$ to $N_s$ do
    Define randomly $\lambda_1$ and $\lambda_2$, such that $\lambda_1+\lambda_2 = 1$;
    $x$ = Constructive-Algorithm($\lambda_1$, $\lambda_2$); //construct $x$ minimizing $\lambda_1 f_1+\lambda_2 f_2$
    $D$ = non-dominated solutions of ($D \cup \{x\}$);
End-for
Main loop:
For $iter = 1$ to $Max\_Iter$ do
    $x$ = select randomly a solutions from $D$;
    Perturbation:
    $x^k$ = remove randomly $k$ medians from $x$ and stored they in $r$; //Destruction
    $D_1 = \{x^k\}$;  // $x^k$ is a partial solution with p-k medians
    For $j = 1$ to $k$ do //Construction loop
        $D_2 = \emptyset$;
        For each partial solution $x^k$ from $D_1$ do
            For each facility $a \notin (x^k \cup r)$ do
                $x' = x^k \cup \{a\}$;  // $x'$ is a solution with p-k+j medians
                Evaluate the partial solution $x'$; //compute $f_1(x')$ and $f_2(x')$
                $D_2$ = non-dominated solutions of ($D_2 \cup \{ x'\}$);
            End-For
            $D_1 = D_2$;
        End-For
    End-For
    $x$ = select randomly a solution from $D_1$;
    $f$ = select randomly an objective function ; // $f = f_1$ or $f = f_2$
    Local Search:
    $D_{LS}$ = Local-Search($x$, $f$); //improve $x$ minimizing  $f$
    Intensification:
    $x^o$ = select randomly a solutions from $D$;
    $x^g$ = select randomly a solutions from $D_{LS}$;
    $D_{PR}$ = Path-Relinking($x^o$, $x^g$);
    $D$ = non-dominated solutions of ($D \cup D_1 \cup D_{LS} \cup D_{PR}$);
End-for
Return $D$;
End BOILS+PR
```

Fig. 1. Pseudocode of the proposed BOILS+PR heuristic

minimum. The proposed algorithm is called Bi-Objective ILS with Path Relinking (denoted by BOILS+PR). The pseudocode description of the algorithm is presented in Figure 1.

The BOILS+PR algorithm has three input parameters: the number of non-dominated solutions built in the initial phase (N_s), a parameter k used in the perturbation phase (k = the number of medians to be removed from a solution) and the maximum number of iterations (*Max_Iter*) used as stop condition. The algorithm returns a set of non-dominated solutions called D. In the next subsections, we described each phase of the BOIL+PR heuristic.

2.1 Initial Solutions

For the p-median problem, a solution $\boldsymbol{x}$ is represented by an array of p numbers that represents the open facilities (or medians). In the first phase of the BOILS algorithm, N_s solutions are constructed using the *sample greedy* algorithm proposed by [13] for the single-objective p-median problem. The set D is initialized by the non-dominated solutions among the Ns constructed solutions (see steps 02-06). In the *sample greedy* algorithm, a solution is constructed trying to minimize the value of the weighted linear utility function $f_\lambda = \lambda_1 f_1 + \lambda_2 f_2$, where λ_1 and λ_2 are nonnegative weights (define randomly) such that $\lambda_1+\lambda_2 = 1$.

2.2 Perturbation

A solution $\boldsymbol{x}$ from the set D is selected to be perturbed. The perturbation procedure is composed of two stages: destruction and construction. In the destruction stage (step 09), k medians are removed from the current solution $\boldsymbol{x}$ and a partial solution $\boldsymbol{x}^k$ (of size p-k) is obtained. The removed medians are stored in $\boldsymbol{r}$. In the construction stage (steps 10-21), from the partial solution $\boldsymbol{x}^k$, a set D_1 of non-dominated solutions (complete solutions) is constructed. In the first step, m-p-k partial solutions (of size p-k+1) are constructed by inserting different location points in $\boldsymbol{x}^k$ (m = number of location points). From these partial solutions, the non-dominated solutions are selected. In the next step, new solutions (of size p-k+2) are obtained by inserting different location points in each of the partial non-dominated solutions. The construction stage ends when the obtained non-dominated solutions are complete (of size p).

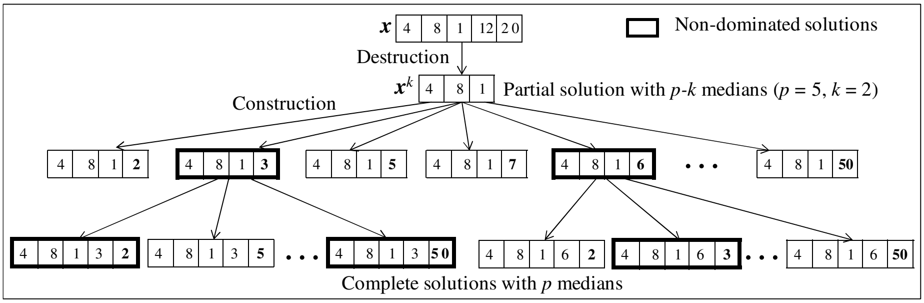


Fig. 2. Solutions constructed by perturbation procedure (example for p=5, k=2 and m=50)

In Figure 2, it is shown an example of the destruction and construction stages. Note that the non-dominated solutions are obtained from $\boldsymbol{x}^k$ by swapping k points.

2.3 Local Search

A solution $\boldsymbol{x}$ selected from D_1 (set of the non-dominated solutions generated in the perturbation phase) is improved by a Local-Search (LS) procedure (steps 22-24). The solution $\boldsymbol{x}$ is improved by making a neighborhood search and minimizing the objective function $f = f_1$ or $f = f_2$ (f is select randomly). Figure 3 shows two examples of the search directions explored by the local search phase.

The neighborhood of a solution $\boldsymbol{x}$ is generated by making swap movements, that is, the neighbor solutions are obtained by swapping location points (or facilities). Given a current solution $\boldsymbol{x}$, the LS procedure determines, for each facility $a \notin s$, which facility $b \in s$ (if any) would improve the solution the most if a and b were interchanged (i.e., if a were opened and b closed). If there is one such improving move, a and b are interchanged. The procedure continues until no improving interchange can be made. We adopt the best improvement strategy, i.e., all neighbors are checked and the best is chosen. The LS procedure returns a set D_{LS} composed by the all the non-dominated solutions found in the LS process.

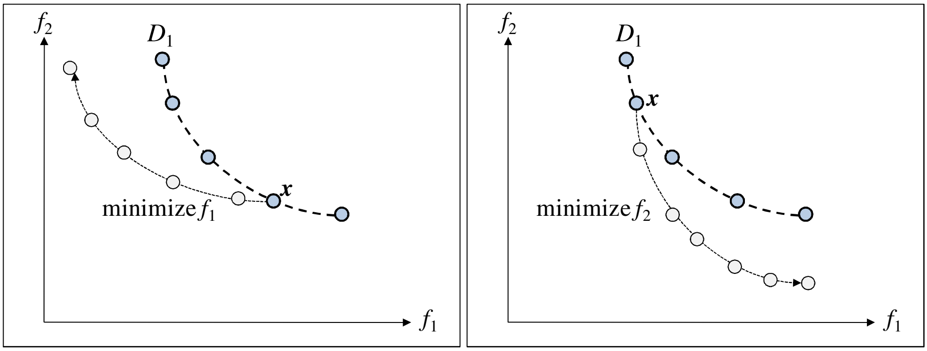


Fig. 3. Examples of search directions explored in the Local Search

2.4 Acceptance Criterion

The acceptance criterion used in the BOILS+PR algorithm is based on the Pareto dominance approach. All the non-dominated solutions found in the LS phase are stored in the set D (step 28). In each iteration of the algorithm, a solution belonging to the set D is selected at random to be perturbed.

A solution $\boldsymbol{x}$ is inserted in the set D if it is not dominated by any solution $\boldsymbol{x}' \in D$, moreover, the solutions dominated by $\boldsymbol{x}$ are removed from the set D.

2.5 Intensification with Path Relinking

In the ILS+PR algorithm we include an intensification procedure (steps 25-27) based on the Path Relinking (PR) technique that explores a trajectory that connect two non-dominated solutions, $\boldsymbol{x}^o$ (origin) and $\boldsymbol{x}^g$ (guide) [5]. Solution $\boldsymbol{x}^o$ is selected from D (the current set of non-dominated solutions) and $\boldsymbol{x}^g$ is selected from D_{LS} (set of non-dominated solutions obtained in the LS phase). The PR procedure starts with $\boldsymbol{x}^o$ and gradually transforms it into the other solution $\boldsymbol{x}^g$ by inserting elements from $\boldsymbol{x}^g - \boldsymbol{x}^o$ and removing elements from $\boldsymbol{x}^o - \boldsymbol{x}^g$. The total number of stages made (to transform $\boldsymbol{x}^o$ into $\boldsymbol{x}^g$) is $| \boldsymbol{x}^g - \boldsymbol{x}^o |$, which is equal to $| \boldsymbol{x}^o - \boldsymbol{x}^g |$ (*symmetric difference* between s_o and s_g). In each stage, the "best" move that reduces the distance from $\boldsymbol{x}^o$ to $\boldsymbol{x}^g$ is executed, that is, the solution obtained by the best move is chosen. That is, in each stage, solutions less similar than the origin solution and more similar than the guide one are obtained. In bi-objective case, the "best" solutions are those which are non-dominated by the all swap neighborhood of the current stage. From the set of non-dominated neighbors, we choose randomly a solution $\boldsymbol{x}$ to be analyzed. In the next stage, new solutions are generated from this solution $\boldsymbol{x}$. The PR procedure ends when $\boldsymbol{x}$ and $\boldsymbol{x}^g$ become equal. The set of the overall non-dominated solutions (set D_{PR}) is kept as the result of the PR procedure.

In Figure 4, it is shown an example in which the guide solution $\boldsymbol{x}^g$ is obtained in three stages from the origin solution $\boldsymbol{x}^o$.

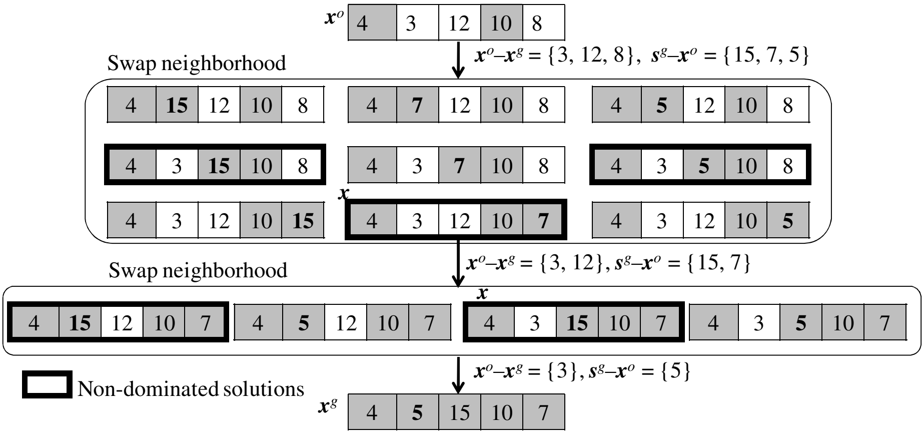


Fig. 4. The Path-Relinking procedure

2.6 Parameters of the Algorithm

We tested the combination of parameters which yielded the best results. *Ns* was adjusted in 20 and *k* was selected at random in the interval [2, min($\lceil p/2 \rceil$, 10)]. The algorithm was executed with *Max_Iter* = 400, 600, 800, 1000 and 1200 iterations. The value that has given certain stability to the algorithm (in terms of quality and number of non-dominated solutions) was *Max_Iter* = 800.

3 The ε-Constraint Algorithm for the BO-*p*-MP

The ε-Constraint algorithm finds Pareto-optimal solutions (*Pareto-optimal frontier*) by solving iteratively the mathematical formulation of the problem with additional constraints [14].

Consider a set J of m location points and a set I of n customers. The Integer Programming formulation of the BO-*p*-MP is as follows:

$$\textit{Minimize} \quad f_1 = \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} \text{ and } f_2 = \sum_{j\in J} c_j y_j \qquad (1)$$

(BO-*p*-MP)

$$\text{subject to: } \sum_{i\in J} x_{ij} = 1, \quad i \in I \qquad (2)$$

$$x_{ij} \le y_j, \quad i \in I, j \in J \qquad (3)$$

$$\sum_{j\in J} y_j = p \qquad (4)$$

$$x_{ij}, y_{\mathrm{j}} \in \{0,1\}, \quad i\in I, j\in J. \qquad (5)$$

The decision variables of the problem are y_{i} and x_{ij}. y_j = 1, if a facility is opened in local $j\in J$, and 0, otherwise; x_{ij} = 1, if customer $i\in I$ is served from the facility located in $j\in J$, and 0, otherwise. The objective functions to be minimized are defined in (1). Constraints (2) guarantee that each customer is attended by one and only one facility. Constraints (3) prevent any customer from being supplied from a local with an unopened facility. The total number of open facilities is set to p by constraint (4). Finally, (5) defines that the decision variables are binary.

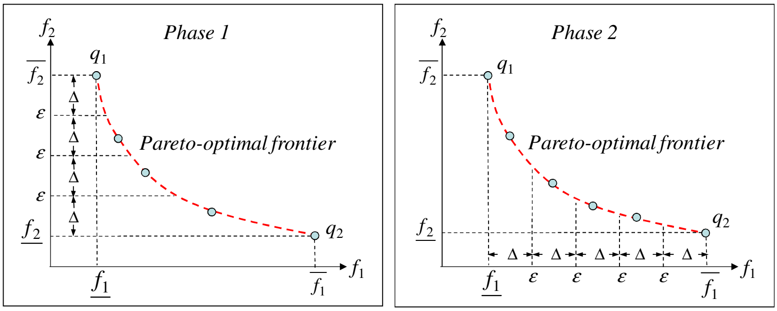


Fig. 5. ε-Constraint algorithm: Phases 1 and 2

The ε-Constraint algorithm has two phases (see Figure 5. Each phase determines a subset of Pareto-optimal solutions.

The first phase starts by finding the extreme point $q_1 = (\underline{f_1}, \overline{f_2})$ of the Pareto-optimal frontier, where:

$\underline{f_1} = Min\, f_1$ subject to: (2), (3), (4) and (5).

$\overline{f_2} = Min\, f_2$ subject to: (2), (3), (4), (5) and $f_1 \leq \underline{f_1}$.

Then, the algorithm determines the minimum value of f_2 ($\underline{f_2}$), solving the model:

$\underline{f_2} = Min\, f_2$ subject to: (2), (3), (4) and (5).

To find a set of *Pareto-optimal* points, the algorithm divides the range $\overline{f_2} - \underline{f_2}$ in s equally sized intervals. The size of each interval is determined by $\Delta = (\overline{f_2} - \underline{f_2})/s$.

The iterative scheme starts with the upper bound $\varepsilon = \overline{f_2} - \Delta$ for f_2 and a new *Pareto-optimal* point $q = (f^*_1, f^*_2)$ is determined solving the following models:

$f^*_1 = Min\, f_1$ subject to: (2), (3), (4), (5) and $f_2 \leq \varepsilon$,
$f^*_2 = Min\, f_2$ subject to: (2), (3), (4), (5) and $f_1 \leq f^*_1$.

Other points are determined considering the upper bounds $\varepsilon = \overline{f_2} - K\Delta$, $K = 2,\ldots,s$. The Phase 1 of the algorithm finishes when $\varepsilon = \underline{f_2}$.

The Phase 2 is a similar and complementary algorithm. It starts with the other extreme point $q_2 = (\overline{f_1} , \underline{f_2})$ of the Pareto-optimal frontier and new Pareto-optimal points $q = (f^*_1, f^*_2)$ are determined solving the following models:

$f^*_2 = Min\, f_2$ subject to: (2), (3), (4), (5) and $f_1 \leq \varepsilon$,
$f^*_1 = Min\, f_1$ subject to: (2), (3), (4), (5) and $f_2 \leq f^*_2$.

The ε-Constraint algorithm does not guarantee finding all the Pareto-optimal solutions. The number of Pareto-optimal solutions depends on the number of intervals (s) used to divide the range of the objectives f_1 and f_2. In each phase of this algorithm we use $s = 50$.

4 Computational Tests

The proposed BOILS+PR algorithm was coded in C++. The ε-Constraint algorithm was coded in Mosel and solved with Xpress-MP. The algorithms were executed on an Intel Core 2 Quad with a 2.4 GHz processor and 3GB of RAM.

In this work, we analyze the behavior of the Path Relinking (PR) technique used in the BOILS algorithm. We compare the non-dominated solutions obtained by BOILS without PR (BOILS) and BOILS with PR (BOILS+PR). The heuristic solutions also are compared with the Pareto-optimal solutions determined by the algorithm ε-Constraint.

4.1 Problem Instances

The ILS+PR algorithm was tested using a set of 150 problem instances. In this work we considered problems in which the number of location points (m) is equal to the number of customers (n). The sizes of the instances are $m \times p$ = (50, 100, 200, 300, 402)×(5, 10, 15, 20, 25, 30). For each combination of m and p, 5 problem instances were generated randomly. The distances d_{ij} from customers i to facilities j were uniformly generated in [0, 3 000]. The cost c_j for installing a facility at the local j is generated uniformly from the interval [1 000, 40 000].

4.2 Performance Measures

The results obtained are evaluated in terms of the number of non-dominated solutions generated by the algorithms (cardinal measure) and the proximity of the obtained non-dominated front to the Pareto front or reference front (distance measures).

We denoted by E the set of Pareto-optimal solutions obtained by ε-Constraint algorithm (Phases 1 and 2). D' and D are the sets of non-dominated solutions obtained by BOILS and BOILS+PR algorithms, respectively. A reference set R is determined. R is formed by the non-dominated solutions obtained by the three algorithms:

$$R = \text{non-dominated solutions of } (E \cup D' \cup D).$$

Note that, $E \subset R$, because set E contains Pareto-optimal solutions. For BOILS and BOILS+PR we compute the number of non-dominated solutions in R: $|D' \cap R|$ and $|D \cap R|$.

We also measure the performance of the set T ($T = E$, D' or D) relative to the reference set R by using the following distance metric [2]:

$$D_{av} = 100 \times \frac{1}{|R|} \sum_{x \in R} \min_{y \in T} d(z^1, z^2) \qquad (6)$$

where, $z^1 = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x})\, f_2(\boldsymbol{x}))$, $z^2 = \boldsymbol{f}(\boldsymbol{y}) = (f_1(\boldsymbol{y})\, f_2(\boldsymbol{y}))$, $d(z^1, z^2) = \max\{(f_1(\boldsymbol{x}) - f_1(\boldsymbol{y}))/\Delta_1$, $((f_2(\boldsymbol{x}) - f_2(\boldsymbol{y}))/\Delta_2\}$ and Δ_j is the range of the objective f_j (j =1,2) in the reference set R.

Note that D_{av} is the average distance from a point $\boldsymbol{f}(\boldsymbol{x}) \in \boldsymbol{f}(R)$ to its closest point in $\boldsymbol{f}(T)$. This D_{av} metric evaluates the distribution of T as well as the proximity of T to R. D_{av} indicator is widely employed measure for multi-objective problems [16][17][18].

4.3 Obtained Results

Table 1 presents the comparison among ε-Constraint, BOILS and BOILS+PR on problem instances with p = 5 and 10 medians. For each set of instances $m \times p$, Table 1 shows the total number of reference solutions provided by each algorithm (on five instances) and the average distance values (D_{av}) for each algorithm. Note that BOILS and BOILS+PR generate their own set of non-dominated solutions (D' and D), which do not necessarily belong to R. For the 50 instances (with p=5 and 10), $|R|$ = 3030 reference solutions were obtained, from which $|E|$ = 2061 (68%) Pareto-optimal solutions were obtained by ε-Constraint algorithm. $|D' \cap R|$ = 2909 (96%) and $|D' \cap R|$ = 2957 (98%) reference solutions were obtained by BOILS and BOILS+PR, respectively. For these 50 problems, the most of the Pareto-optimal solutions (obtained by ε-Constraint algorithm) were determined by BOILS and BOILS+PR. For example, the BOILS+PR algorithm found 2013 Pareto-optimal solutions (97.6%).

From Table 1, we can observe that the BOILS and BOILS+PR algorithms find a larger number of reference solutions. Also, It can be seen that the solutions generated by BOILS and BOILS+PR algorithm are closer to the reference solutions (see D_{av}). Considering the cardinal and distance measures, the performance of BOILS+PR was slightly superior to BOILS on problem instances with p = 5 and 10 medians.

Table 1. Performance of the algorithms for instances with p = 5 and 10 medians

Instances	**Reference**	**ε-Constraint**		**BOILS**		**BOILS+PR**	
$m \times p$	$\|R\|$	$\|E\|$	D_{av}	$\|R\cap D'\|$	D_{av}	$\|R\cap D\|$	D_{av}
50×5	151	136	0.037	151	0.00	151	0.00
100×5	144	123	0.031	142	0.004	**143**	**0.002**
200×5	163	138	0.045	163	0.00	163	0.00
300×5	181	152	0.034	180	0.002	**181**	**0.00**
402×5	282	203	0.075	281	0.001	**282**	**0.00**
Total	921	752	0.222	917	0.007	**920**	**0.002**
50×10	288	215	0.063	288	0.00	288	0.00
100×10	313	229	0.063	312	0.0008	**313**	**0.00**
200×10	441	290	0.072	430	0.004	**432**	**0.001**
300×10	508	293	0.078	476	0.015	**485**	**0.008**
402×10	559	282	0.081	486	0.018	**519**	**0.014**
Total	2109	1309	0.357	1992	0.037	**2037**	**0.024**

Table 2 shows the mean computational times (in seconds) spent by the three algorithms on problems with p = 5 and 10. The ε-Constraint algorithm, to solve an problem of size 402×10, spends about 15 hours in average. For problems with $p \geq 15$ and m = 402, the computational time of this algorithm is very expensive and it was impossible to generate a set of Pareto-optimal solutions. The BOILS+PR algorithm, to solve an instance of size 402×10, spends 380 seconds in average.

Table 2. Computational for instances with p = 5 and 10 medians

Instances	**ε-Constraint**	**BOILS**	**BOILS+PR**
$m \times p$	*Time* (*seconds*)	*Time* (*seconds*)	*Time* (*seconds*)
50×5	54.5	0.8	0.9
100×5	163.8	3.0	3.8
200×5	1662,1	14.6	16.4
300×5	10086.1	36.6	34.2
402×5	41362.7	73.0	77.6
Total	53329.2	128.0	132.9
50×10	128.4	2.0	3.2
100×10	299.3	13.8	15.6
200×10	2674.9	75.6	78.2
300×10	13314.4	190.4	198.7
402×10	54856.3	373.8	380.5
Total	71273.3	655.6	676.2

In Table 3 we present the results obtained by the algorithms BOILS and BOILS+PR on problem instances with p = 15, 20, 25 and 30 medians. For these problems, we can observe that the number of reference non-dominated solutions increases as p increases (except for problems with m = 50). For all problem sizes, except for 50×25 and 50×30, BOILS+PR generates more reference solutions, when compared to BOILS. For the 100 instances (with p =15, 20, 25 and 30), $|R|$ = 20850 reference solutions were obtained, from which $|D'\cap R|$ = 15485 (74,2%) and $|D\cap R|$ = 18196 (87,2%) solutions were obtained by BOILS and BOILS+PR, respectively.

Table 3 also shows that BOILS+PR performs better than the BOILS algorithm regarding the distance measure. The computational times spent by BOILS+PR are slightly greater than the times of BOILS.

In all problems tested we observed that the objectives distance (f_1) and cost (f_2) are conflicting. A solution with minimum value for f_1 has a high value for f_2 and vice versa. Figure 6 shows this fact.

Table 3. Performance of the algorithms for instances with p = 15, 20, 25 and 30 medians

Instances	**Reference**	**BOILS**			**BOILS+PR**		
$m \times p$	$\|R\|$	$\|R\cap D'\|$	D_{av}	*Time* (*s*)	$\|R\cap D\|$	D_{av}	*Time* (*s*)
50×15	364	354	0.0018	2.2	**363**	**0.0006**	2.4
100×15	600	574	0.01	18.6	**593**	**0.001**	20.8
200×15	709	552	0.088	120.8	**659**	**0.011**	126.4
300×15	753	544	0.06	373.4	**679**	**0.023**	385.5
402×15	901	651	0.065	732.8	**752**	**0.038**	741.3
Total	3327	2675	0.2248	1247.8	**3046**	**0.0736**	1276.4
50×20	394	387	0.0013	2.6	**394**	**0.00**	3.4
100×20	847	801	0.0068	32.2	**835**	**0.0013**	39.5
200×20	978	727	0.053	213.4	**883**	**0.0121**	219.6
300×20	1177	818	0.056	680.5	**1034**	**0.117**	691.2
402×20	1263	839	0.068	1088.2	**1067**	**0.029**	1095.5
Total	4659	3572	0.1851	2016.9	**4213**	**0.1594**	2049.2
50×25	360	360	0.00	3.4	360	0.00	3.9
100×25	1076	991	0.01	58.4	**1014**	**0.006**	64.2
200×25	1422	1059	0.048	461.2	**1201**	**0.022**	473.8
300×25	1505	1038	0.063	974.5	**1209**	**0.029**	986.2
402×25	1522	970	0.068	1698	**1254**	**0.036**	1703.4
Total	5885	4418	0.189	3195.5	**5038**	**0.093**	3231.5
50×30	309	309	0.00	3.6	309	0.00	4.4
100×30	1151	1014	0.012	80.3	**1100**	**0.004**	87.8
200×30	1739	1250	0.043	601.6	**1405**	**0.023**	612.3
300×30	1969	1270	0.057	1305.5	**1591**	**0.031**	1342.2
402×30	1811	977	0.075	2202	**1494**	**0.02**	2235.5
Total	6979	4820	0.187	4193	**5899**	**0.078**	4282.2

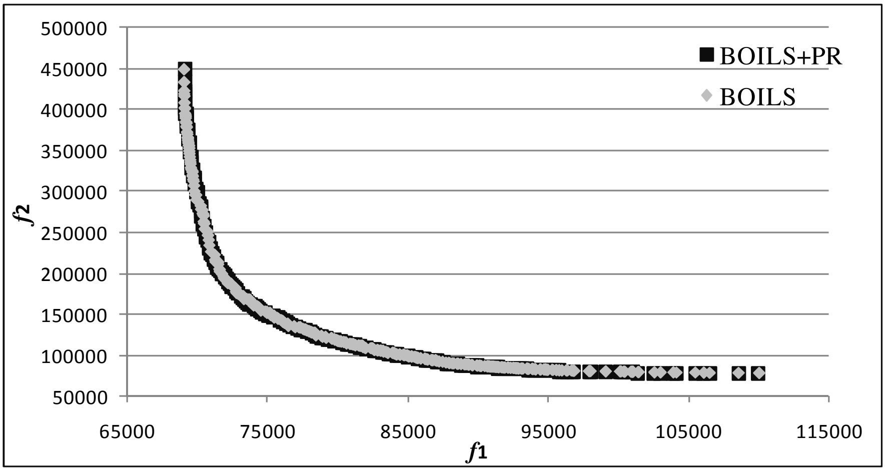


Fig. 6. Non-dominated points determined by BOILS and BOILS+PR for a problem with $m = 300$ and $p = 30$

5 Conclusions

In this paper we propose an ILS heuristic with Path-Relinking to solve a bi-objective *p*-median problem. The goal of the proposed heuristic is to produce close to Pareto-optimal solutions. For instances with up to 402 location points and 10 medians, a subset of Pareto-optimal solutions were determined by a Mathematical Programming algorithm.

In this work we show that, the combination of the ILS and Path-Relinking provides a useful tool to solve multi-objective optimization problems. The proposed approach is simple and proved to be very effective. For a total of 150 bi-objective problems tested, 23880 reference (non-dominated) solutions were obtained, from which 18394 solutions were obtained by BOILS and 21153 solutions were obtained by BOILS+PR (2759 more).

In the literature, the ILS heuristic was little applied to multi-objective problems. It would be very interesting to apply the proposed algorithm to other types of combinatorial problems.

Acknowledgment

This work was funded by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and the Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG). We also thank the *Dash Optimization* for providing full license of Xpress-MP.

References

1. Arroyo, J.E.C., Vieira, P.S., Vianna, D.S.: A GRASP algorithm for the multi-criteria minimum spanning tree problem. Ann. Oper. Res. 159, 125–133 (2008)
2. Czyzak, P., Jaszkiewicz, A.: Pareto simulated annealing - a metaheuristic technique for multiple objective optimization. J. Multi-Crit Decis. Making 7, 34–47 (1998)

3. Gandibleux, X., Ehrgott, M.: 1984-2004 – 20 Years of Multiobjective Metaheuristics. But What About the Solution of Combinatorial Problems with Multiple Objectives? In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 33–46. Springer, Heidelberg (2005)
4. Geiger, M.J.: Improvements for multi-objective flow shop scheduling by Pareto Iterated Local Search. In: 8th Metaheuristics International Conference, pp. 195.1–195.10 (2009)
5. Glover, F.: Tabu search and adaptive memory programming - advances, applications, and challenges. In: Interfaces in Computer Science and Operations Research, pp. 1–75. Kluwer Academic Publishers, Dordrecht (1996)
6. Jones, D.F., Mirrazavi, S.K., Tamiz, M.: Multi-objective meta-heuristics: An overview of the current state-of-art. Eur. J. Oper. Res. 137, 1–19 (2002)
7. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems. II. The p-medians. SIAM J. on Applied Mathematics 37, 539–560 (1979)
8. Liang, Y.-C., Lo, M.-H.: Multi-objective redundancy allocation optimization using a Variable Neighborhood Search algorithm. J. Heuristics 16(3), 511–535 (2010)
9. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search. In: Glover, F. (ed.) Kochenberger Handbook of metaheuristics, pp. 321–353. Kluwer Academic, Dordrecht (2003)
10. Medaglia, A.L., Villegas, J.G., Rodrígues-Coca, D.M.: Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. J. of Heuristic 15(2), 153–176 (2009)
11. Mladenovic´, N., Brimberg, J., Hansen, P., Moreno-Pérez, J.A.: The p-median problem: A survey of metaheuristic approaches. Eur. J. Oper. Res. Research 179, 927–939 (2007)
12. Reese, J.: Solution methods for the p-median problem: An annotated bibliography. Networks 48(3), 125–142 (2006)
13. Resende, M.G.C., Werneck, R.F.: A hybrid multistart heuristic for the uncapacitated facility location problem. Eur. J. Oper. Res. 174(1), 54–68 (2005)
14. Villegas, J.G., Palacios, F., Medaglia, A.L.: Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. Ann. Oper. Res. 147(1), 109–141 (2006)
15. Yagmahan, B., Yenisey, M.M.: Ant colony optimization for multi-objective flow shop scheduling problem. Comp. Ind. Eng. 54(3), 411–420 (2008)
16. Framinan, J.M., Leisten, R.: A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. OR Spectrum 30(4), 787–804 (2008)
17. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Trans. Evol. Comput. 7(2), 204–223 (2003)
18. Al-Fawzana, M.A., Haouari, M.: A bi-objective model for robust resource-constrained project scheduling. Int. J. Prod. Econ. 96, 175–187 (2005)
19. Arroyo, J.E.C., dos Santos, P.M., Soares, M.S., Santos, A.G.: A Multi-Objective Genetic Algorithm with Path Relinking for the p-Median Problem. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS, vol. 6433, pp. 70–79. Springer, Heidelberg (2010)
20. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2006)

A Framework for Locating Logistic Facilities with Multi-Criteria Decision Analysis

Gilberto Montibeller[1] and Hugo Yoshizaki[2]

[1] Management Science Group, Dept. of Management, London School of Economics, England, United Kingdom
[2] Dept. of Production Engineering, University of Sao Paulo, Brazil

Abstract. Locating logistic facilities, such as plants and distribution centres, in an optimal way, is a crucial decision for manufacturers, particularly those that are operating in large developing countries which are experiencing a process of fast economic change. Traditionally, such decisions have been supported by optimising network models, which search for the configuration with the minimum total cost. In practice, other intangible factors, which add or reduce value to a potential configuration, are also important in the location choice. We suggest in this paper an alternative way to analyse such problems, which combines the value from the topology of a network (such as total cost or resilience) with the value of its discrete nodes (such as specific benefits of a particular location). In this framework, the focus is on optimising the overall logistic value of the network. We conclude the paper by discussing how evolutionary multi-objective methods could be used for such analyses.

Keywords: multi-criteria analysis, logistics, facility location, multi-attribute value theory, multi-objective optimisation.

1 Introduction

Designing logistic networks – involving plants, distribution centres and cross dock terminals – are strategic decisions for industrial companies (Daskin 1995, Ballou 2004, Klose and Drexl 2005, Melo et al. 2009). Locating such logistic facilities in an optimal way is a crucial and frequent decision for manufacturers (Bowersox et al. 2007), in particular for those companies operating in large developing countries which are experiencing a process of fast economic change.

Traditionally, location decisions have been modelled as a network with discrete location alternatives. These models can then be optimised to find the configuration with the minimum total cost. In practice, other intangible factors, which add or reduce value to a potential configuration, are also important in the location choice (Daskin 1995, Klose and Drexl 2005). However, these factors are many times taken into account just exogenously during the analysis.

We suggest in this paper an alternative way to analyse such problems, which combines the value from the topology of a network (such as total cost) with the value of its discrete nodes (such as specific benefits of a particular location). In

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 505–519, 2011.

this framework, the focus is on optimising the overall logistic value of the network, considering the criteria that contribute to adding value to the system and the preferences of the company involved in the decision. The framework is presented using an illustrative case, based on the authors' experience in supporting this type of decisions in Brazil.

The paper has the following structure. We start describing the main features present in logistic facility location decision problems and the decision models usually employed for analysing them. We then present the proposed framework. The paper concludes with suggestions for further research on the topic, in particular the use of evolutionary multi-objective methods in this context.

2 The Logistic Facility Location Problem

Decisions involving logistic network design have some specific challenges, which make it extremely difficult to make an informed decision without mathematical modelling and decision support. These challenges are:

- *Systemic properties* : Multi-location decision problems have intrinsic systemic properties, where each topology provides a set of different performances. Examples of systemic properties may be total cost, geographical covering, or resilience. This requires the use of optimisation modelling for analysing the problem.
- *Properties of Elements* : Additionally, nodes in location decision problems have properties that distinguish one from each other. For example, potential sites for an industrial plant could have different levels of performance, such as skilled manpower availability or transportation infrastructure. This would lead to a discrete choice analysis of nodes, without the use of optimisation tools.
- *Multiple Objectives* : When companies are considering location problems, they have a set of objectives they want to achieve. These objectives may reflect concerns about systemic properties of the network (increase profitability, improve coverage, etc.) as well as properties of its elements (availability of skilled manpower, quality of local infra-structure, etc.).
- *Preferences and value trade-offs* : When more than one criterion is involved in the decision, for instance the need to minimise costs versus the wish to have wider coverage (and thus expensive) topology, then there is the need of modelling the company's preferences and trade-offs (e.g. costs versus coverage).
- *Facilitated decision modelling* : The decision process needs to be carefully crafted, as it should allow a consistent and participative decision making process (Franco and Montibeller 2010), where managers are able to negotiate their preferences and trade-offs. It thus should enable decision-makers to "play" with the models (de Geus 1988), assessing the consequences of different topologies and trade-offs and, therefore, an interactive decision tool is required.

We now contrast these problem's features with the existing literature on supporting facility location, which is the subject of the following section.

3 Decision Models for Facility Location Problems

In this section we review first the traditional mono-criterion decision models for facility location suggested in the literature, followed by models that allow the consideration of multiple objectives.

3.1 Traditional Decision Models

Facility location has a long history in the scientific tradition of mathematical modelling (see reviews by Klose and Drexl 2005, and Melo et al. 2009). From this literature, one can conclude that most classical location models in logistics and supply chain management have a single objective function, generally focused on the minimisation of costs (or other surrogate criterion, such as total weighted distance or number of open nodes).

On the other hand, some of those authors caution that location problems in logistics are definitely multi-objective decisions. For instance, Daskin (1995) comments that non-quantifiable objectives and other issues will influence sitting decisions to a great extent, and solutions of a single objective model are optimal in a narrow sense. Klose and Drexl (2005) state that strategic decisions, such as location of logistic facilities, are often multi-objective in nature, and, according to them, the body of literature regarding multiple criteria location models is very limited.

Evidently, in real world interventions, one can construct useful decision location models based on a single criterion, then conduct extensive changes to the model to try to include multiple objective issues (for example, adding a constraint which expresses a minimum level of achievements of a given objective). However, this may prevent decision makers to contemplate radically different, but high-value, topologies. This suggests the use of multiple criteria models for location decisions, which is reviewed briefly below.

3.2 Decision Models Considering Multiple Objectives

The recognition that facility location decisions have an inherent multi-objective nature has led to the development of several approaches for incorporating multiple criteria in the decision models (see Current et al. 1990, Malczewski and Ogryczak 1996 and Nickel et al. 2005).

Here we confine ourselves in briefly describing such models according to a categorisation of benefits which we are proposing. Within this perspective, there are two main types of decision models that incorporate multiple criteria:

- *Topological Benefits* : The more traditional way of incorporating multi-criteria into facility location models is with the inclusion of topological metrics, a subset of the network's systemic properties. Such metrics attempt to reflect the decision-makers' concerns about the topology of the network (e.g. distance between nodes, service coverage, among others). Models including topological benefits employ either goal programming (see Tamiz et al. 1998) or multi-criteria optimisation (see Marler and Arora 2004). Applications using the former approach are, for example, Badri et al. (1998) and Giannikos (1998); and using the latter approach are, for instance, Hugo and Pistikopoulos (2005) and Yang et al. (2007).

- *Nodal Benefits* : Another way of analysing facility location problems is by considering the several benefit dimensions, for each potential site of the network, which the decision-makers are concerned with (e.g., level of infrastructure, availability of labour, among others). Each of these nodes has an intrinsic level of benefits and disadvantages that need to be taken into account in the decision, which are properties of the elements of the network. This type of evaluation can be easily analysed by multi-criteria discrete alternative methods (see Wallenius et al. 2008), where each node is an alternative in the model (e.g. Keeney 1979, Min 1994, Yurimoto and Matsui 1995).

Considering only one type of benefit may be detrimental to the analysis, in our opinion. Methods which include topological benefits lack an evaluation of benefits at the node level and are incapable of dealing with intangible benefits. On the other hand, methods which evaluate nodal benefits do not consider the network structure and, therefore, the benefits that some particular topological layouts may provide. Given these concerns, it is rather surprising that there are a limited number of suggested approaches which try to assess both types of benefits, such as Badri (1999), Cheng et al. (2003), and Farahani and Asgari (2007).

However, none of these approaches mentioned in the former paragraph recognised explicitly the measurement of two distinctive types of benefits, as we are suggesting in this paper. Also, when trying to assess topological benefits, they assumed linear marginal value functions, but this is not always a realistic assumption (Stewart 1996). Furthermore, they did not recognise the importance of facilitated decision modelling when implementing such models in practice. In the next section we are proposing a framework for analysing facility location problems which addresses these issues.

4 A Framework for Using Multi-Criteria Analysis in Facility Location Problems

4.1 The Traditional Optimisation Model

We will consider a multi-criteria, single commodity, capacitated facility location problem (SCFL'), but the framework could be employed for other similar problems. It will be illustrated by a case study inspired on real applications in the food and retail industry in Brazil which we supported, as consultants, in the past. In these problems, a manufacturer has to choose the sites for a number of plants, each one with limited production capacity, to fulfill product demand in different markets (client regions). Traditionally this problem could be analysed using a mono-criterion model (see Aikens 1985, Klose and Drexl 2005):

$$\text{Objective function: } \min C_L = \sum_{i \in I} \sum_{j \in J} c_{ij}\, x_{ij} \; + \sum_{i \in I} f_i\, z_i \tag{1}$$

Subject to:

$$\sum_{j \in J} x_{ij} \leq S_i z_i \,, \forall i \in I \tag{2}$$

$$\sum_{i \in I} x_{ij} = D_j \,, \forall j \in J \tag{3}$$

$$x_{ij} \geq 0 \,, \forall\, i,\, j \tag{4}$$

$$z_i \in \{0, 1\} \tag{5}$$

Where: x_{ij} are decision variables (flow between plant i and client j); z_i are decision variable (1, if facility i is established, 0 otherwise); c_{ij} are unit production and distribution cost associated with satisfying client j from plant i; f_i is the fixed cost of plant i; D_j is the demand of client region j; S_i is the capacity of plant i; I,J are sets of candidate sites for plants and clients, respectively. Equation (1) is the mono-criterion objective function, which minimises the total logistic cost (production and distribution) plus fixed costs associated with selected alternatives for a plant site. The constraint set (2) prevents that open plants suffer from violations of their capacities (upper bounds). The constraint set (3) assures that all demand from all clients will be satisfied, while the remaining constraints are the usual non-negativity conditions (4) and binary variable definitions (5).

In our illustration, a food company has to design its logistic network (location of plants and capacity allocation to clients). There are ten alternatives (potential sites) to place factories, chosen from main State capital cities (see Table 1, based on the Brazilian market). Their capacity (3,820 t/year), fixed costs (861,000 US$/year, including investment and overhead for a given production capacity) and variable costs (55.88 US$/t) are assumed to be the same for every alternative. The demand is split between 23 Brazilian States (first column of Table 1) easily accessed by land transportation and concentrated in the State capitals. The demand (last column of Table 1) is calculated as the product of per capita consumption times the State population.

Transportation costs to hauling products from a given factory to a given client region are calculated from actual freight rates and are also shown in Table 1. The optimal solution $c_L{}^*$will find which sites will be opened and, therefore, their location and number, in order to minimise (1). In the following sections we present how this model can be altered in order to consider both nodal and topological benefits.

4.2 Identifying and Measuring Nodal Benefits

A nodal benefit is here defined as a benefit inherent to a specific node and, thus, is concerned about a property of the network's elements. It may be either a tangible or intangible aspect, and nodes (potential sites) may be assessed by any number of criteria. There are several ways of identifying nodal benefits. The analyst could use the existing location literature and select the ones suitable for the particular problem. Or else, the analyst could relate to similar problems and see which benefits were measured in these case studies. A third option is to define tailor-made indices from the client company's objectives. This latter approach is the one that we favour, as it links clearly the company's strategic objectives with the fundamental objectives in locating plants (see Keeney 1992). For example, in the illustrative case study mentioned in the previous section, four nodal benefits could be defined, which would reflect the strategic objectives of the company: *maintenance efficiency*, *planning permission*, *logistic services* and *skilled labour*.

Table 1. Freight Rate from City to State and Demand of Each State

	Cities - Potential Sites										
State	JOV	CUR	SAP	RIO	BHR (US$/t)	GOI	SAL	REC	FOR	SOL	Demand (t)
01) AL	115.32	113.18	98.25	86.81	76.19	85.64	32.69	20.65	43.71	66.28	256.0
02) BA	94.29	92.11	77.22	66.31	55.14	66.59	21.24	41.70	57.37	70.60	1221.0
......	......	......	......	......	......	......	......	......	......	......	......
22) SE	106.41	104.22	89.33	78.21	67.26	77.03	23.77	29.60	48.08	66.40	160.0
23) TO	100.51	96.18	87.22	87.83	72.66	52.41	65.45	79.08	66.66	48.14	104.00

Whatever the method employed to define the set of nodal benefits, the next step is to define an attribute (i.e. a performance index) and assess a value function over its range. For example, in our illustration, the nodal benefit *maintenance efficiency* of a potential city could be measured by the following performance index: 'number of hours required from a breakdown to a full repair' with an associated value function (see Figure 1). For nodal benefits that have a qualitative nature, discrete labels can be defined to represent a given level of performance, as shown in Table 2 for measuring the *logistic services* benefit (for details see Keeney 1992).

If there are several nodal benefits that are preferentially independent of each other (see Keeney 1992), then they can be aggregated using a simple weighted sum. Thus the overall nodal benefit for a given topology can be calculated by:

$$\nu_N = \sum_{p \in P} w_{N_p} \nu_{N_p} \tag{6}$$

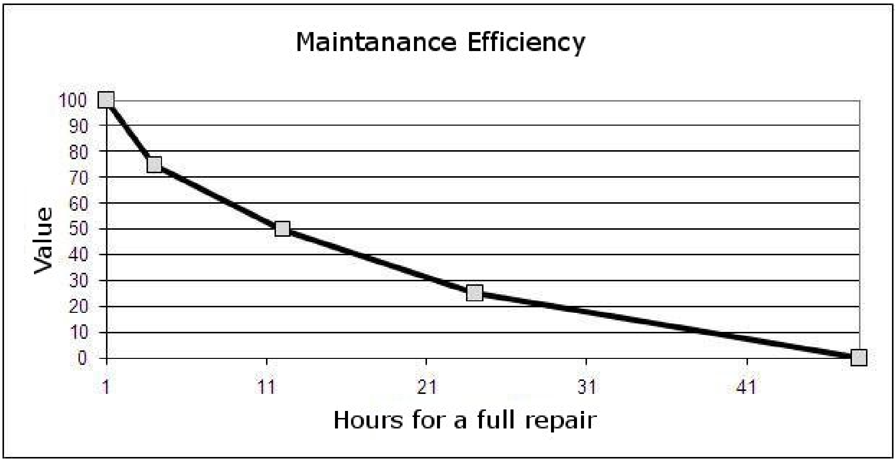


Fig. 1. Marginal value for nodal quantitative benefits - Maintenance Efficiency

Table 2. Marginal value for nodal qualitative benefits - Logistic Services

Value	Description of attribute level
100	Availability of mid-size carriers with a national coverage and high services standards.
75	Availability of mid-size carriers with a regional coverage and high services standards.
50	Few large-size carriers with a national coverage and average service standards.
25	Availability of small-size carriers with regional coverage with high services standards.
0	Few small-size carriers with regional coverage with average services standards.

Where v_{Np} is the partial value on the p-th nodal benefit for this given topology; w_{Np} is the weight of the p-th nodal benefit; P is the set of nodal benefits and $\sum w_{Np} = 1$. Notice that v_N will depend on the number of nodes active, thus:

$$\nu_N = \sum_{p \in P} w_{N_p} \nu_{N_p} = \sum_{p \in P} w_{N_p} \left[\frac{\sum_{i \in I} \nu_{ip} z_i}{\sum_{i \in I} z_i} \right] \tag{7}$$

Equation (7) creates a non-linearity in the objective function. In this paper, we opted for using a direct enumeration method, so a conventional mixed integer linear programming software could be readily used.

4.3 Identifying and Measuring Topological Benefits

Topological benefits are evaluation criteria that assess systemic properties from the network configuration as a whole. Classical mono-criterion location models optimise topological benefits, such as minimising total costs or the weighted distances from clients (a service level surrogate). The same type of criteria is employed in many multi-objective location models, as listed in Current et al.

(1990). In our framework we suggest considering total costs against overall benefits (which can be either topological or nodal), as the former are extremely important in logistic network design. In addition to total cost and geographical coverage, other topological benefits may be employed, such as: number of sites, service area overlap, total system risk, total environmental impact, total quality of service (Badri et al. 1998, Giannikos 1998, Hugo and Pistikopoulus 2005, Farahani and Asgari 2007).

Most methods for measuring topological benefits proposed in the literature use direct measurement of performance (a rare exception is Mirchandani and Reilly 1987) but, again, from a decision analytic point of view, we believe it is important that non-linear value functions are represented in the model, in a similar way as shown in Figure 1.

If there is more than one topological benefit and if these benefits are preferentially independent, then they also can be aggregated using a simple weighted sum. Therefore the overall topological benefit for a given topology can be calculated by:

$$\nu_T = \sum_{m \in M} w_{T_m} \nu_{T_m} \tag{8}$$

Where v_{Tm} is the partial value on the m-th topological benefit for this given topology; w_{Tm} is the weight of the m-th topological benefit; M is the set of topological benefits and $\sum w_{Tm} = 1$.

4.4 Measuring Preferences for Costs

In the same way as it was done for benefits, we suggest that a value function should be elicited for total logistic costs. Given the large amount of resources typically required in this type of investment, one would expect a non-linear value function, as increases from the minimum cost should be heavily penalised. In practice, we can find the minimum total cost $c_L{}^*$, disregarding the benefits, and then calculate the ratio $c_L =$ cost of the layout/minimum total cost. A value function can then normalise this attribute $v_L = f(c_L)$ (the higher the ratio, the less valuable the solution is, in terms of its overall cost, as shown in Figure 2).

In the illustrative case, component costs are fixed plant costs and variable logistic costs, the latter involving transportation and handling (Table 1). The total logistic costs are calculated by (1). Raw material and manufacturing costs are assumed to be the same for every potential site and thus were not included in the model.

4.5 Defining the Overall Logistic Value Optimisation Model

In our framework, instead of optimising the total cost, as formulated in (1), we suggest a model that maximises the overall logistic value of the network. Thus Eq. 1 is replaced by:

$$Max\, V = w_L \nu_L + w_B [w_N \nu_N + w_T \nu_T] \tag{9}$$

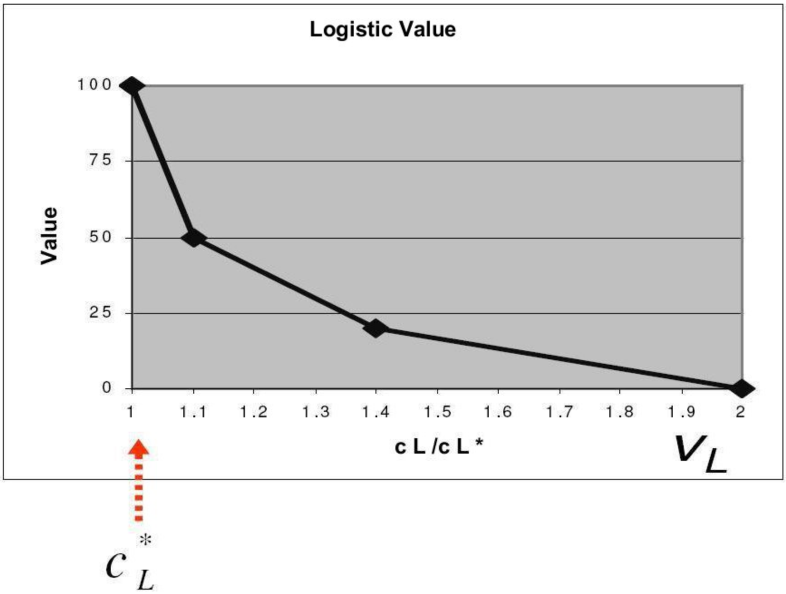


Fig. 2. A value function for total logistic cost

Where V is the overall logistic value of a given topology, w_L is the weight for the logistic cost criterion and w_B is the weight for the logistic benefits (with $w_L + w_B = 1$ and $w_N + w_T = 1$).

4.6 Determining Value Trade-Offs

The next step in our framework is to determine the trade-offs between logistic costs, topological benefits and nodal benefits, represented in (9) by their weights. Defining trade-offs is a crucial step in any important decision involving multiple-objectives, but there are many common mistakes in eliciting them (Keeney 2002), which can lead to meaningless values. For a multi-attribute value model, as the one we are proposing here, trade-offs must be elicited taking into account the ranges of each performance index (for example, in Figure 1, from 1 to 48 hours). Traditionally, trade-offs in multi-attribute value functions are elicited *a priori* from the client, often using the swing weighting method (see Keeney 2002).

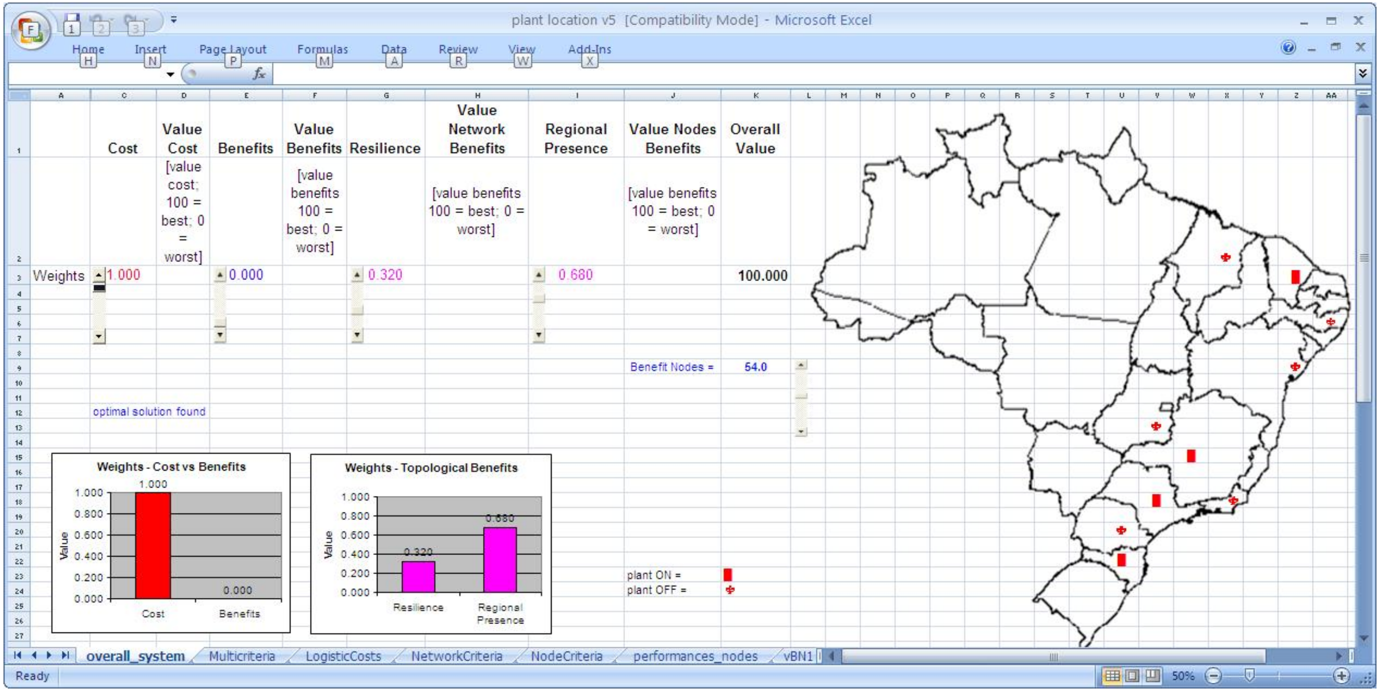
We suggest that weights are elicited in phases, i.e., intra-weights w_{N_p} for nodal criteria, intra-weights w_{T_m} for topological criteria, then inter-weights w_N and w_T for nodal and topological benefits, followed by w_L and w_B for cost and overall benefits. In this way, the analyst can reduce the cognitive burden involved in the elicitation (see also von Winterfeld 1999) and help decision-makers think explicitly about the nodal *versus* topological trade-offs, as well as about cost *versus* overall benefits trade-off. Another possibility is to elicit all the intra-criteria weights at once and then infer the inter-weights for benefits from the swing weights allocated to the bottom-level benefit criteria. In any case, once the trade-offs are defined, Equation 9 can be solved to find the configuration which provide the highest overall value.

If the analyst wants to explore different alternatives without *a priori* preference information, for example ranging w_L against w_B in (9), a diagram of efficient solutions can be drawn. Notice, however, that this method does not find all efficient solutions for non-convex fronts (see Marler and Arora 2004). Furthermore, this type of analysis may struggle to convey the results when more than two criteria are considered. That is why we suggest the importance of an interactive visual decision support system in this type of analysis, which is described next.

4.7 Exploring the Solution Layouts

In our experience in supporting this type of decision, location problems require several interactions between analyst and client for exploring alternative solutions. Typically, decision makers want to see an optimal (minimum cost) logistic solution first, in order to anchor their bottom line expectations and start their search for other solutions that provide better benefits within acceptable cost levels. There is, however, an understandable reluctance to explore options which are far from the optimal one, or topologies which are radically different than the optimal one. Furthermore, any analysis that exogenously considers benefits, relying on ad hoc what-if trials, is obviously an inefficient way to explore the solution space. Multi-criteria analysis, on the other hand, allows a more comprehensive evaluation of the benefits in the solution space. It searches for high value solutions, considering the trade-offs between benefits and costs. In this type of approach, visual interactive modelling becomes crucial - the role of the model is to explore different configurations and see how the trade-offs would impact on the topology of the network.

In order to illustrate how such interaction between model and decision-maker may be performed in practice, we have implemented the model described here as an Excel-based decision support system. Changing the weights of cost and benefits leads the model to present a new optimal topology, which is depicted geographically in the map. Figure 3, for example, shows the case where all emphasis is placed on total logistic costs: the network topology can be easily seen on the map, with the different weights shown in the bar graphs. Four sites are opened, appearing as rectangles in the Brazilian map; potential but unused sites are marked as crosses. Different priorities would lead to different layouts, for instance, if all weight were thrown at the benefits then nine plants would be opened. The latter solution maximises network benefits and the topology is therefore quite different than the own shown in Figure 3.



A	C	D	E	F	G	H	I	J	K
	Cost	Value Cost	Benefits	Value Benefits	Resilience	Value Network Benefits	Regional Presence	Value Nodes Benefits	Overall Value
		[value cost; 100 = best; 0 = worst]		[value benefits 100 = best; 0 = worst]		[value benefits 100 = best; 0 = worst]		[value benefits 100 = best; 0 = worst]	
Weights	1.000		0.000		0.320		0.680		100.000
								Benefit Nodes =	54.0
	optimal solution found								
								plant ON =	🟥
								plant OFF =	✚

Fig. 3. A visual interactive model for analysing logistic value - All weight on cost

5 Conclusions and Directions for Further Research

This paper proposed a framework for locating logistic facilities with optimisation and multi-criteria value analysis. This was illustrated by an example, based on real data of food companies in Brazil, where a multi-objective single commodity plant location model was developed and implemented as an interactive decision support system. Such an approach may be relevant in complex business environments, such as developing economies, where other considerations beyond total costs are crucial in facilities location (for instance, the availability of infrastructure, technical personnel, logistic services and industrial utilities).

One of the main potential contributions of this paper is to suggest that these issues can be better analysed with a proper categorisation of benefits into topological (network) and nodal (site) types, measured and evaluated in a rigorous way, using multi-attribute value theory. While there is a large literature on multi-criteria facility location, the approach we are suggesting is the first one – as far as we know – that recognises the distinctive role of these benefits and suggests an integrated way of assessing the potential solutions of the network when such benefits are considered. Another potentially relevant contribution is that our framework stresses, within a network optimisation perspective, the relevance of measuring the marginal value of performances and the importance of a proper elicitation of weights in multi-criteria facility location models. Finally, the interactive use of a decision-support system to guide the appraisal of solution, while not new in multi-criteria analysis, is hardly seen in this kind of logistic analysis.

As any decision method, the framework we suggested does have limitations. First, there is a need for the analyst to specify the benefits and elicit value functions and weights for nodal benefits. Second, it is heavy in terms of the computational time required, which could make it less suitable for larger or more complex models. Third, it creates a non-linear objective functions when nodal benefits are considered (i.e. Equation 7). Fourth, it is unable to find solutions in non-convex regions of the Pareto front. There are, therefore, several avenues of further research, as we suggest below.

Non-linearity in the objective function and computational speed. The method we suggested in the paper (running the model for a given number of open plants and finding the number which maximises the overall value, i.e., enumeration) is simple, albeit computationally time consuming. Thus research into ways of making it faster, or solving it directly with the non-linear objective function would be welcomed. Another issue to be investigated is how to find efficient solutions which are not located in the convex front.

We believe that the use of evolutionary methods, as well as other multi-objective meta-heuristics methods (Jones et al. 2002), is an interesting avenue of research for coping with these challenges. There are several levels in which such approaches could be employed. At the first level, one could use a suitable mono-criterion evolutionary method to solve (9), given its non-linear nature.

At the second, and more interesting, level one could employ Multi-Objective (EMO) methods (Deb 2001) for this type of problem. For instance, they could be used to find efficient solutions on the bi-dimensional space v_L versus $w_N \nu_N + w_T \nu_T$ in Equation 9. At a third level, researchers could explore how to deal with the multiple (topological and nodal) benefits of this problem, using EMO methods.

Recent developments in this field, with the incorporation of preferences into EMO methods, as suggested for example by Branke and Deb (2005) and Koksalan and Phelps (2007), may provide an alternative to a full specification of preferences *a priori*. Researchers using EMO could then investigate what is the best way to display the solutions, when multiple dimensions are considered (e.g., performances on multiple topological benefits). They also could identify what type of preference should be provided and how such preferences could be elicited in a decision support system, in a way that is user-friendly and also more efficiently supports decision making. Another avenue of research is using other formulations in (9), for finding efficient solutions located in non-convex regions of the Pareto front, such as the Tchebycheff norms (see Marler and Arora 2004).

Decision Support Systems (DSSs) with visual interaction for logistics. The use of DSSs that have a visual interface with the user and which allow them to "play with the model" (de Geus 1988) could be extended to other decisions in logistics, such as procurement of logistic service operators. More crucially, there is a need for further research into which types of visual interaction are more helpful in these contexts. While such a visual interaction is relatively common now for multi-criteria decision analysis software supporting the evaluation of discrete alternatives, as well as discrete-event simulation software (see Belton and Elder 1994), it seems an aspect under-developed in multi-criteria analysis for logistic problems.

Decision conferencing for facility location decisions. The application of this framework in real business cases would rely on model-based facilitation (Franco and Montibeller 2010) and decision support tools. Research could be conducted on the social outcomes of such interventions (for example, commitment to action, satisfaction of users, etc.). Another issue is how the benefit criteria is structured for a particular problem, for example how concerns about fairness of supply or other social issues could be incorporated in the model.

Concluding, we hope that this paper may further stimulate research on the links between optimisation of logistic problems and multi-criteria decision analysis, particularly with a multi-attribute value perspective. In this context, there are opportunities to use evolutionary multi-objective algorithms, and some open avenues of research were suggested here. We believe that this is a rich area for research and that it may support real-world logistic decisions in a more comprehensive way.

References

Aikens, C.H.: Facility location models for distribution planning. European Journal of Operational Research 22, 263–279 (1985)

Badri, M.A.: Combining the analytic hierarchy process and goal programming for global facility location-allocation problem. International Journal of Production Economics 62, 237–248 (1999)

Badri, M.A., Mortagy, A.K., Alsayed, A.: A multi-objective model for locating fire stations. European Journal of Operational Research 110, 243–260 (1998)

Ballou, R.H.: Business Logistics/Supply Chain Management. Prentice-Hall, New Jersey (2004)

Belton, V., Elder, M.D.: Decision support systems: Learning from visual interactive modelling. Decision Support Systems 12, 355–364 (1994)

Bowersox, D.J., Closs, D.J., Cooper, M.B.: Supply Chain Logistics Management, 2nd edn. McGraw-Hill, New York (2007)

Branke, J., Deb, K.: Integrating user preferences into evolutionary multi-objective optimisation. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, pp. 461–478. Springer, Berlin (2005)

Cheng, S., Chan, C.W., Huang, G.H.: An integrated multi-criteria decision analysis and inexact mixed integer linear programming approach for solid waste management. Engineering Applications of Artificial Intelligence 16, 543–554 (2003)

Current, J., Min, H., Schilling, D.: Multiobjective analysis of facility location decisions. European Journal of Operational Research 49, 295–307 (1990)

Daskin, M.S.: Network and discrete location: models, algorithms, and applications. Wiley, New York (1995)

Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, New York (2001)

de Geus, A.P.: Planning as Learning. Harvard Business Review, 70–74 (March-April 1988)

Farahani, R.Z., Asgari, N.: Combination of MCDM and covering techniques in a hierarchical model for facility location: A case study. European Journal of Operational Research 176, 1839–1858 (2007)

Franco, A., Montibeller, G.: Invited Review - Facilitated modeling in Operational Research. European Journal of Operational Research 205, 489–500 (2010)

Giannikos, I.: A multiobjective programming model for locating treatment sites and routing hazardous wastes. European Journal of Operational Research 104, 333–342 (1998)

Hugo, A., Pistikopoulos, E.N.: Environmentally conscious long-range planning and design of supply chain networks. Journal of Cleaner Production 13, 1471–1491 (2005)

Jones, D.F., Mirrazavi, S.K., Tamiz, M.: Multi-objective meta-heuristics: An overview of the current state-of-the-art. European Journal of Operational Research 137, 1–9 (2002)

Keeney, R.L.: Evaluation of Proposed Storage Sites. Operations Research 27, 48–64 (1979)

Keeney, R.L.: Value-Focused Thinking. Harvard University Press, Cambridge (1992)

Keeney, R.L.: Common mistakes in making value trade-offs. Operations Research 50, 935–945 (2002)

Klose, A., Drexl, A.: Facility location models for distribution system design. European Journal of Operational Research 162, 4–29 (2005)

Koksalan, M., Phelps, S.: An Evolutionary Metaheuristic for Approximating Preference-Nondominated Solutions. INFORMS J. on Computing 19, 291–301 (2007)

Malczewski, J., Ogryczak, W.: The multiple criteria location problem: 2. Preference-based techniques and interactive decision support. Environment and Planning A 28, 69–98 (1996)

Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidisc. Optim. 26, 369–395 (2004)

Min, H.: Location analysis of international consolidation terminals using the Analytic Hierarchy Process. Journal Of Business Logistics 15, 25–44 (1994)

Mirchandani, P.B., Reilly, J.M.: Spatial distribution design for fire fighting units. In: Ghosh, A., Rushton, G. (eds.) Spatial Analysis and Location-Allocation Models, pp. 186–223. Van Nostrand Reinhold, New York (1987)

Melo, M.T., Nickel, S., Saldanha-da-Gama, F.: Facility location and supply chain management - a review. European Journal of Operational Research 196, 401–412 (2009)

Nickel, S., Puerto, J., Rodriguez-Chia, A.M.: MCDM location problems. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple criteria decision making: state of the art surveys, pp. 761–795. Springer, Berlin (2005)

Stewart, T.J.: Robustness of Additive Value Function Methods in MCDM. Journal of Multi-Criteria Decision Analysis 5, 301–309 (1996)

Tamiz, M., Jones, D., Romero, C.: Goal programming for decision making: An overview of the current state-of-the-art. European Journal of Operational Research 111, 569–581 (1998)

von Winterfeld, D.: On the relevance of behavioral decision research for decision analysis. In: Shanteau, S., Mellers, B.A., Schum, D.A. (eds.) Decision science and technology: reflections on the contributions of Ward Edwards, pp. 133–154. Kluwer, Norwell (1999)

Wallenius, J., Dyer, J.S., Fishburn, P.C., Steuer, R.E., Zionts, S., Deb, K.: Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead. Management Science 54, 1336–1349 (2008)

Yang, L., Jones, B.F., Yang, S.-H.: A fuzzy multi-objective programming for optimization of fire station locations through genetic algorithms. European Journal of Operational Research 181, 903–915 (2007)

Yurimoto, S., Masui, T.: Design of a decision support system for overseas plant location in the EC. Int. J. Production Economics 41, 411–418 (1995)

Lorenz versus Pareto Dominance in a Single Machine Scheduling Problem with Rejection

Atefeh Moghaddam*, Farouk Yalaoui, and Lionel Amodeo

ICD, LOSI, University of Technology of Troyes,
12 rue Marie Curie, 10010 Troyes, France
{atefeh.moghaddam,farouk.yalaoui,lionel.amodeo}@utt.fr

Abstract. Scheduling problems have been studied from many years ago. Most of the papers which were published in this domain are different in one or many of issues as following: objective functions, machine environment, constraints and methodology for solving the problems. In this paper we address the problem of single machine scheduling in which due to some constraints like capacity, rejection of a set of jobs is accepted. The problem is considered as bi-objective one: minimization of the sum of weighted completion times for the accepted jobs and minimization of the sum of penalties for the rejected jobs. We find that in this problem, the solutions are not handled in a satisfactory way by general Pareto-dominance rule, so we suggest the application of Lorenz-dominance to an adapted bi-objective simulated annealing algorithm. Finally we justify the use of Lorenz-dominance as a useful refinement of Pareto-dominance by comparing the results.

Keywords: Scheduling, rejection, Pareto-dominance, Lorenz-dominance, bi-objective simulated annealing.

1 Introduction

In most classical scheduling problems, it is assumed that all jobs should be processed on the pre-defined machines and rejection of jobs is not permitted. However, in real world situation, the case is more complicated. Due to different reasons we may ignore some jobs in the schedule. For example, we have limited capacity for the available machines so we cannot process all jobs on the machines. Or due to the long processing times of some jobs, we prefer outsourcing and asking other factories to do the processing on the jobs while paying the cost. Even in some cases, tardiness in not permitted so if we cannot meet the due dates of the jobs, it is better to reject those jobs from the beginning of the scheduling process in order to avoid paying more cost. In all mentioned cases, we can conclude that there are some situations in which we have to reject some jobs in order to gain more profit (even we reject those jobs forever and we pay the cost of loosing customers or we prefer to outsource the jobs and pay the related fees).

* Corresponding author.

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 520–534, 2011.

Here, we have a system with a single machine and n jobs $J_1, J_2, ..., J_n$ in an off-line environment. Job J_j $(j = 1, ..., n)$ has a weight value w_j, processing time p_j and rejection cost r_j. At each time, the machine can process only a single job and the preemption is not allowed, so when we start processing a job, we cannot do any interruption. Subset S of the n jobs is the set of accepted jobs and $\bar{S}$ the set of rejected jobs. We have two objective functions: (1) minimization of total weighted completion time of accepted jobs (S), (2) minimization of total penalty of rejected jobs $(\bar{S})$. We have calculated the correlation between these two objectives by generating many instances randomly. The average correlation shows that these two objective functions are not correlated and we can deal with this problem as a bi-objective one. Our goal is to choose the solutions which balance these two criteria.

It is obvious that we cannot minimize both objective functions simultaneously so we should find a compromise between these two criteria. In most of previous works in which rejection of jobs was considered, the researchers have added the rejection costs to the main criterion and considered the problem as a single objective one and discussed the computational complexity of the problem so given approximation or competitive algorithms.

In most of the researches in which rejection has been studied, minimization of summation of makespan and rejection cost in different machine environment has been considered [1,2,3,4,5,6,7,8]. For the first time, Bartal et al. [1] introduced the notion of rejection and considered the problem of minimizing the makespan plus the sum of penalties on m parallel identical machines in on-line and off-line setting. Seidan [2] modified the problem of Bartal et al. In his study, the preemption was allowed and the situation was on-line. By accepting preemption, he could reach to the better performance of the system. Hoogeveen et al. [3] studied the problem with unrelated parallel machines while preemption was allowed. Lu et al. [4] considered the unbounded parallel batch machine scheduling with release dates in off-line setting. Zhang et al. [5] studied the same problem as Lu et al, but when there was single machine. Cheng and Sun [6] studied single machine scheduling problems with different objective functions as makespan, total weighted completion times and max lateness/tardiness in which the processing time of the jobs was a linear function of its starting time. Lu et al. [7] considered the bounded single-machine parallel-batch scheduling problem in off-line setting. In [8], Cao and Yang studied a model of parallel batch scheduling where jobs arrived dynamically.

There are two papers, in which another decision variable as machine purchasing was presented in the model. In those problems, the machine purchasing cost was added to the makespan and rejection cost. Dósa and He [9] and Nagy-György and Imreh [10] considered when a job is arrived, the decision maker has the following alternatives: reject it, non-preemptively process it on an existing machine or purchase a new machine and assign it to this machine.

Bansal et al. [11] studied minimization of flow time and job idle time while rejection is allowed.

Engels et al. [12], considered the summation of the same objective functions in this paper, therefore treated the problem as a single objective one, and they denoted the overall optimality criterion as $\sum_{j\epsilon S} w_j C_j + \sum_{j\epsilon \overline{S}} r_j$. It has been shown that adding the option of rejection makes the problem $1||\sum_{j\epsilon S} w_j C_j + \sum_{j\epsilon \overline{S}} r_j$ weakly *NP*-Complete so it cannot admit polynomial-time solutions unless $P{=}NP$[12]. A comprehensive literature review with more details is available on [13].

To the best of our knowledge, there are very few papers in this area, in which this problem has been considered as a bi-objective one. Jolai et al. [14] developed two metaheuristic methods, one based on Pareto-simulated annealing and another based on colonial competitive algorithm. They showed that both algorithms are capable enough to find good estimation of the Pareto-optimal set. Moghaddam et al. [13] proposed a bi-objective simulated annealing algorithm to solve this problem and compare the results with the exact Pareto-optimal solutions. In [15], they proposed a mathematical model and used this model for implementing Two-Phase Method (TPM) [16] as an exact method to find all Pareto-optimal solutions (supported and non-supported ones); and finally in [17], they adapted three different bi-objective simulated annealing algorithms to this problem and proposed the best one.

In this paper both objective functions are considered separately so we try to find a set of solutions. Pareto-dominance rule is a classical approach for solving multi-objective optimization problems in which a set of Pareto non-dominated solutions are found. But in some multi-criteria decision problems, the Pareto-optimal solution concept is not powerful enough, so another approach called Lorenz-dominance has been introduced. As an example, Dugardin et al. [18] showed that in reentrant scheduling problem, the L-NSGA2 which works based on Lorenz-dominance can achieve more interesting results by comparing to NSGA-2 (based on Pareto-dominance).

However, in all the researches cited above, the concept of Pareto-dominance was considered. In this paper, due to the huge number of (estimated) Pareto-optimal solutions found in [17], we apply Lorenz-dominance in order to decrease the number of choices and facilitate the selection process for decision maker while improving even the solutions found by the best algorithm proposed in [17].

This paper is organized as follows. Part 2 presents Pareto and Lorenz dominance properties. In part 3, we describe implementation of TPM for finding exact Pareto-optimal solutions. Bi-objective simulated annealing algorithm based on Lorenz-dominance properties is presented in part 4. Numerical results are reported and discussed in part 5, and the last part provides our conclusions.

2 Pareto and Lorenz Dominance Properties

We remind that a feasible schedule S is Pareto-optimal, or non-dominated, with respect to the performance criteria f_1 and f_2 in a minimization problem, if there is no feasible schedule S' such that both $f_1(S') \leq f_1(S)$ and $f_2(S') \leq f_2(S)$, where at least one of the inequalities is strict. Fig. 1 shows the Pareto-dominance area of a solution S.

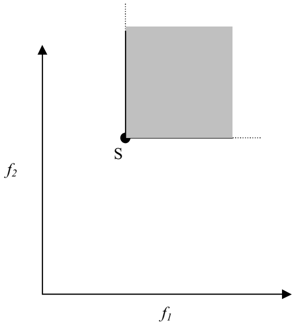


Fig. 1. Pareto domination structure

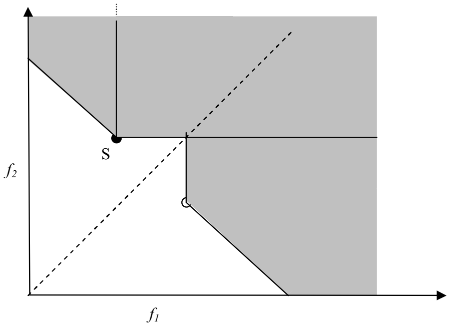


Fig. 2. Lorenz domination structure

In some multi-objective combinatorial optimization problems, the set of Pareto-optimal solutions can be huge, so decision maker cannot evaluate all choices easily. To deal with this problem, application of Lorenz-dominance would be a solution. The Lorenz-optimal set is a subset of the Pareto-optimal set so it reduces the size of the non-dominated front as it is shown in Fig. 2. On the other hand, when we use meta-heuristics, Lorenz-dominance may help us to find better estimation of Pareto-solution than Pareto-dominance, as the search space in Lorenz is more restricted. This fact has been shown in Dugardin et al. [18]. Here, we define the Lorenz-dominance as below:

Definition: For two objectives f_1 and f_2, Lorenz vector for the solution S is:

$L(S) = (f_{(1)}, f_{(1)} + f_{(2)})$
Where $f_{(1)} \geq f_{(2)}$ which means: $f_{(1)} = \max(f_1, f_2)$

In other words, for calculating Lorenz vector, first we should sort the value of objective functions in non-increasing order, we call it sorted set. Then the first element of Lorenz vector with be the first entry in sorted set, the second element will be the summation of first two entries in sorted set, ...

Solution S dominates solution S' in Lorenz context if Lorenz vector of S, shown by $(L(S))$, dominates Lorenz vector of S', denoted by $(L(S'))$, in Pareto context.

$\forall S, S'; \quad S <_L S' \Leftrightarrow L(S) <_P L(S')$

For review of Lorenz-dominance concept, readers may refer to [19,20,21].

3 Two-Phase Method (TPM) Implementation

TPM was proposed for the first time by Ulungu and Teghem for solving a bi-objective assignment problem [16]. The objective of TPM is to find all Pareto-optimal solutions. In that paper, they introduce two kinds of efficient solutions:

supported (SS) and non-supported (NSS). In the first phase, they find all SS by aggregating two objective functions z_1 and z_2 to the form of $\lambda_1 z_1 + \lambda_2 z_2$. Information from SS identified by phase one is used to reduce the search space for computing the NSS. In phase two, each triangle defined by two consecutive SS is explored separately to find all NSS. The objective of TPM is to find all Pareto-optimal solutions.

Remark: For each job we have processing time (p_j), weight value (w_j) and rejection cost (r_j). In the rest of this part, it is assumed that the jobs are sorted and re-numbered in non-decreasing order of $\frac{p_j}{w_j}$ ratio. In other words, job 1 is the job with smallest value of $\frac{p_j}{w_j}$ positioned in the first place, and so on.

3.1 Phase One Implementation

The two objective functions can be formulated as follows:

$$\text{Min } z_1 = \sum_{j \in S} w_j C_j \ . \tag{1}$$

$$\text{Min } z_2 = \sum_{j \in \overline{S}} r_j \ . \tag{2}$$

Where:
S : set of accepted jobs
$\overline{S}$: set of rejected jobs
w_j : weight value of job j
C_j : completion time of job j
r_j : penalty of rejecting job j

In following steps, we find a complete set of SS:

Step 1: We find two initial solutions by minimizing each of the objective functions separately. Suppose u is the solution from minimizing z_1 and v is calculated by minimizing z_2.

Step 2: Let $\lambda_1 = z_2(u) - z_2(v)$ and $\lambda_2 = z_1(v) - z_1(u)$.

Step 3: In this step we find optimum solution for $Z = \lambda_1 z_1 + \lambda_2 z_2$. For this purpose we formulate the model as below [22]:

Model(1):

$$\text{Min } Z = \lambda_1 \left[\sum_{j=2}^{n} \sum_{i=1}^{j-1} (w_j p_i x_j x_i) + \sum_{j=1}^{n} (w_j p_j x_j) \right] + \lambda_2 \sum_{j=1}^{n} (r_j (1 - x_j)) \ . \tag{3}$$

where $x_j = \begin{cases} 0, & \text{if job } j \text{ is rejected;} \\ 1, & \text{if job } j \text{ is accepted.} \end{cases}$

p_j : processing time of job j

We remind that job j is the job placed in position j after being sorted by $\frac{p_j}{w_j}$ ratio. As we re-numbered the jobs, the job no. and job position will be the same.

In order to make the model linear, we apply Mccormick's envelope rule and we get the new model as below. For more details regarding the linearization and the constraints added below, readers may refer to [23]:

Model (2):

$$\text{Min } Z = \lambda_1 \left[\sum_{j=2}^{n} \sum_{i=1}^{j-1} (w_j p_i h_{ij}) + \sum_{j=1}^{n} (w_j p_j x_j) \right] + \lambda_2 \sum_{j=1}^{n} (r_j (1 - x_j)) \; . \tag{4}$$

where

$$h_{ij} \leq x_i \; . \tag{5}$$

$$h_{ij} \leq x_j \; . \tag{6}$$

$$h_{ij} \geq x_i + x_j - 1 \; . \tag{7}$$

$x_i \in \{0, 1\} \; \forall i : 1, 2, ..., n - 1$
$x_j \in \{0, 1\} \; \forall j : 2, 3, ..., n$
$h_{ij} \in \{0, 1\}$
p_j : processing time of job j

By solving Model (2), we find a solution named ss_1.

Step 4: We search between (u, ss_1) and (ss_1, v) for finding new SS by repeating steps 2 and 3. We continue searching up to the time no new SS will be found.

At the end of this phase, we will have a complete set of SS.

3.2 Phase Two Implementation

After finding all supported solutions, in this phase all the triangles underlying each pair of adjacent SS are explored in order to find all NSS. In the case of minimization, such solutions necessarily belong to the interior of the right hand side triangle defined by two consecutive SS [16]. By the following steps, a complete set of NSS between two adjacent supported solutions t_n and t_{n+1} are found. In model (3), we assume that $z_1(t_n) \leq z_1(t_{n+1})$ so before working on each pair of supported solutions, we sort them in non-decreasing order of z_1.

Step 1: In second phase we use again Model(2) but we have to limit the search space to the defined triangles by adding two more constraints (constraints (12) and (13)). Model (3) shows this fact.

Model (3):

$$\text{Min } Z = \lambda_1 \left[\sum_{j=2}^{n} \sum_{i=1}^{j-1} (w_j p_i h_{ij}) + \sum_{j=1}^{n} (w_j p_j x_j) \right] + \lambda_2 \sum_{j=1}^{n} (r_j (1 - x_j)) \ . \tag{8}$$

where

$$h_{ij} \leq x_i \ . \tag{9}$$

$$h_{ij} \leq x_j \ . \tag{10}$$

$$x_i + x_j - h_{ij} \leq 1 \ . \tag{11}$$

$$\sum_{j=2}^{n} \sum_{i=1}^{j-1} (w_j p_i h_{ij}) + \sum_{j=1}^{n} (w_j p_j x_j) \leq z_1(t_{n+1}) \ . \tag{12}$$

$$\sum_{j=1}^{n} (r_j (1 - x_j)) \leq z_2(t_n) \ . \tag{13}$$

$x_i \in \{0,1\}$ $\forall i : 1, 2, ..., n-1$
$x_j \in \{0,1\}$ $\forall j : 2, 3, ..., n$
$h_{ij} \in \{0,1\}$
p_j : processing time of job j

Step 2: After finding a NSS from the first step (called ns_i), we search the area between the NSS and the SS with minimum z_1 (here called t_n) and we find another NSS. We repeat this step until no new NSS is found. (Fig. 3)

Step 3: In this step we start finding a NSS between the first NSS found in step 1 (called ns_i) and t_{n+1}. We find another NSS. Always we search between the last NSS found in this step and the SS which minimized second objective function. We repeat this step until no new NSS is found. (Fig. 3)

Step 4: Here, we find all the NSS which are located between two adjacent NSS found in previous steps. We stop searching when no new solution is found. In this case, the final set would be the complete set of NSS between two supposed consecutive SS. (Fig. 4)

In this method, each time that we search for a new solution, we solve a binary integer model by using branch and bound. When the number of jobs increases, the computational time causes problem. Also we find that the number of exact Pareto-optimal solutions increases, so in next part we adapt a bi-objective simulated annealing algorithm to find a set of estimated solutions which are non-dominated in Lorenz context in order to decrease both computational time (comparing to the exact method) and number of solutions.

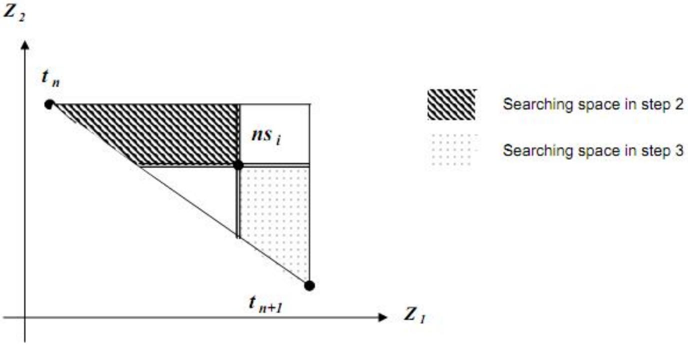


Fig. 3. Searching space for finding non-supported solutions in steps 2 and 3

4 Bi-objective Simulated Annealing Algorithm Based on Lorenz-Dominance Properties

Simulated annealing (SA), as it is named, is coming from the simulation of annealing process [24]. The temperature reduces in each step. At each temperature the system is perturbed and the energy level in calculated. If the energy level has decreased, the perturbed system is considered as the current state if not, the new state is accepted with a probability calculated according to the acceptance probability function. In this method, acceptance probability function has been defined for the aim of accepting the worse solutions in order to have diversification in searching process. However, the probability of accepting worse solutions in high temperature is much more than in low temperature.

The goal of all multi-objective meta-heuristic algorithms is to find a set of Pareto-optimal solutions which is a good estimation of exact solutions set. Over past years, many multi-objective simulated annealing algorithms have been proposed while their differences are mostly in acceptance probability functions. Moghaddam et al. [17], [22] have adapted MOSA proposed in [25,26,27] called MOSA(I), MOSA(II) and MOSA(III) respectively to our problem and concluded that the set of solutions found by MOSA(III) is the best estimation of Pareto-optimal solutions by comparing to the two other algorithms. But as it has been shown in [17] and also table 1, by increasing the number of jobs, the number of Pareto-optimal solutions increases significantly so evaluation and comparison of the potential solutions become more complex for decision maker therefore we

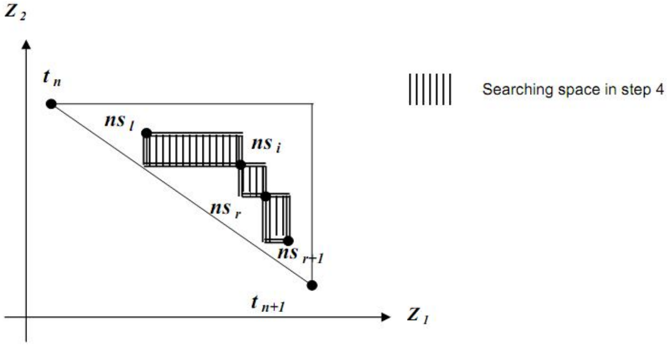


Fig. 4. Searching space for finding non-supported solutions in step 4 [22]

find that in our problem, refinement of Pareto-optimal solutions is inevitable, so we apply Lorenz-dominance to the MOSA in order to overcome this drawback and we call it here in after L-MOSA. L-MOSA works in detail as below:

Step 1: The matrix of jobs' specifications as processing time (p_j), weight value (w_j) and rejection cost (r_j) is created as input data. Then the jobs are sorted by non-decreasing ratios of $\frac{p_j}{w_j}$.

Step 2: We initialize the set of Lorenz non-dominated solutions. For this purpose, we create a matrix with $n + 6$ columns, where n is the number of jobs. The first n columns in each row represent the accepted jobs with their order and "zero" represents the rejected jobs, column $n + 1$ and $n + 2$ represents the value of each objective functions (total weighted completion time for accepted jobs and total penalty for rejected jobs respectively) then the value of column $n + 1$ and $n + 2$ are scaled and reflected in column $n + 3$ and $n + 4$. We scale the value of both objective functions in order to have a better distribution of solutions if not, based on Lorenz properties, the solutions would be converged near to one of the objective functions. We have tested many different scaling functions and finally we found that if we scale column $n+1$ and $n+2$ in a way that both initial Pareto-optimal solutions (shown by u and v in sub-section 3.1, step 1) would be selected after applying the Lorenz-dominance, we can reach to the good solutions. Column $n + 5$ and $n + 6$ filled with the maximum value and the summation of column $n+3$ and $n+4$ respectively (Lorenz property). So we initialize the solution matrix by minimizing each objective functions separately. These solutions are given as the starting points to the L-MOSA.

Step 3: We select one of the initial solutions and consider it as the current solution. Then we start searching for another solution in neighborhood of current solution. The neighborhood structure is as follows: we generate a random integer number from [1,n-1] where n is the number of jobs. We call it a. Then we choose a number of jobs randomly. If each of these a jobs is in the set of accepted jobs in current solution, we reject it and vice versa. In this way, we generate a new solution in neighborhood of the current solution. More detail of this structure has been described in Moghaddam et al. [13].

Step 4: After finding another solution, we should decide whether to accept it as current solution or not. A wide diversified set of weights is considered. Based on the weights, summation of weighted objective values is calculated for the candidate solution and compared with the current solution. If the candidate solution is better than the current one, automatically we accept it and replace it with current solution, if not, we calculate the acceptance probability function as defined in [27].

Step 5: By generating the new solution, this solution will be candidate to enter into the Lorenz non-dominated solutions set. If this solution is dominated by any of the current solutions in this set, then it will not be allowed for entering to the set. If this solution dominates some of the solutions which are already in the set, this solution will be entered to the set and also eliminates the dominated solutions. If the candidate solution dominates none of the current solutions, it will enter to the set of estimated Lorenz non-dominated solutions. By defining different weight values for objective functions in each iteration, one set of Lorenz-non dominated solutions is found and finally, those sets are integrated for achieving final set.

5 Numerical Experiments

5.1 Parameters Setting

We generate several problems randomly within predefined intervals as follows: Processing time, p_j: random integer number within [10,80]; Weight value, w_j: random integer number within [1,30]; Rejection cost, r_j: $\exp(5+\sqrt{a} \times b)$, where a is a random integer number within [1,80] and b is a random number within [0,1].

In order to use the simulated-annealing algorithm, we need to specify the following parameters: initial temperature, epoch length (number of iterations at a given temperature), cooling rate, final temperature and termination criterion.

Here, for L-MOSA we use the same values for parameters presented in [17] in order to be able to compare the results with MOSA. So we fix the initial temperature on 300, the cooling rate as 0.9 and the epoch length as 100. The search process is terminated when T falls below 10. For distributing uniformly the weight values for both objective functions, we divide the interval [0,1] to 10.

5.2 Numerical Results

For evaluating the performance of L-MOSA and MOSA, we generate several problems randomly as described above.

For small size problems, up to 15 jobs, MOSA and TPM find the same Pareto-optimal solutions. When we apply Lorenz-dominance to TPM, we find the same solutions by TPM and L-MOSA. As it is shown in table 1, the number of solutions after Lorenz-dominance implementation has been reduced from 30 to 12 in average.

Table 1. Comparing TPM and L-MOSA results for small size problems (for each no. of jobs, three random problems generated and the averages are reported)

No. of Jobs	No. of Solutions		Computational Time		
	Pareto-Dominance TPM=MOSA	Lorenz-Dominance TPM=L-MOSA	TPM	MOSA	L-MOSA
5	11	4	0.68	6.19	5.48
7	18	9	4.74	7.51	6.29
9	25	8	42.45	8.32	6.01
11	27	18	151.56	8.28	5.52
13	37	14	2522.94	8.73	6.37
15	62	16	15718.50	9.66	6.29
Average	30	12	3073.48	8.12	5.99

In Fig. 5, we show how the number of solutions is decreased by applying Lorenz-dominance properties for a problem with thirteen jobs.

For large size problems, where exact method is unable to find set of exact solutions in reasonable time, we compare the performance of L-MOSA with MOSA. For doing such evaluation, we use several metrics to be able to consider all aspects i.e. convergence, diversity and running time: number of solutions, computational time, ratio of solutions found by an algorithm which are dominated by at least one solution found by another algorithm (c) introduced by Zitzler et al. [28] and $\bar{\mu}_d$ introduced by Riise [29] and modified by Dugardin et al. [18] in which the distance between each solution in one front to its orthogonal projection on other front is measured. In table 2 the results have been shown. The

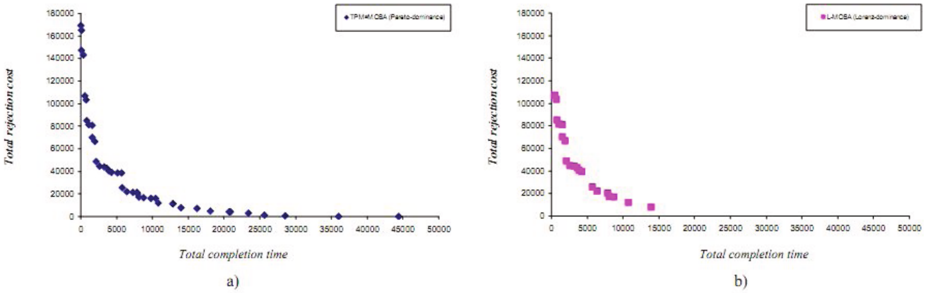


Fig. 5. For thirteen jobs, the solutions found by a) TPM and MOSA based on Pareto-dominance and b) L-MOSA based on Lorenz-dominance properties

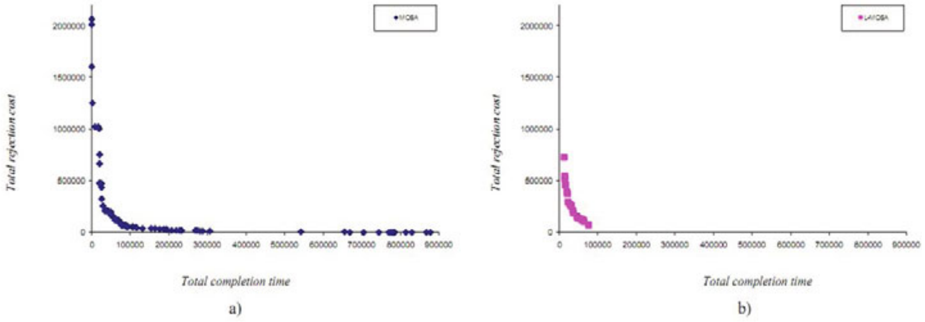


Fig. 6. For sixty jobs, the solutions found by a) MOSA based on Pareto-dominance and b) L-MOSA based on Lorenz-dominance properties

first column is the number of jobs, the second and third, number of solutions found by MOSA and L-MOSA, column 4 and 5, show the computational time for each algorithm, in column 6 and 7, the ratio of solutions found by MOSA (L-MOSA) and dominated by at least one solution found by L-MOSA (MOSA) are presented. The last column shows the distance between two fronts generated by MOSA and L-MOSA.

Table 2. Comparing MOSA and L-MOSA results for large size problems (for each no. of jobs, three random problems generated and the averages are reported) * I=MOSA, II=L-MOSA

No. of jobs	No. of solutions		Computational time		$C_{I,II}$	$C_{II,I}$	$\overline{\mu}_{dI,II}$
	I^*	II^*	I	II			
20	106	23	10.24	6.74	0.0126	0.0000	0.0000
25	109	28	9.92	6.36	0.0127	0.0588	0.0001
30	115	44	10.08	8.03	0.0574	0.1042	0.0001
35	115	37	9.93	7.08	0.1106	0.0824	-0.0003
40	113	29	10.37	7.79	0.1061	0.3209	0.0003
45	105	21	10.26	7.56	0.1238	0.2301	-0.0001
50	108	24	10.71	9.53	0.1455	0.1510	-0.0011
60	93	18	10.56	10.38	0.2928	0.2242	-0.0044
70	92	13	10.85	10.60	0.1171	0.4160	-0.0003
80	83	19	11.26	11.79	0.1151	0.2669	-0.0020
90	91	14	11.42	12.04	0.1190	0.6322	0.0008
100	80	12	12.33	12.88	0.1556	0.3038	-0.0020
125	82	13	13.34	15.54	0.2096	0.2388	-0.0035
150	92	10	15.00	16.43	0.1009	0.5053	0.0003
175	80	15	16.66	20.02	0.1249	0.2735	-0.0042
200	87	12	18.33	23.63	0.1205	0.2803	-0.0037
Average	97	21	11.95	11.65	0.1203	0.2555	-0.0013

As it is shown in the last row of table 2, the number of solutions found by L-MOSA is about one fifth of the number of solutions found by MOSA. The computational time for both algorithms is approximately the same. In average, 12% of the solutions found by MOSA are dominated by L-MOSA solutions while 25% of the solutions found by L-MOSA are dominated by MOSA. The average distance between these two front is negative which means that the L-MOSA front in below the MOSA front.

In Fig.6, the solutions found by MOSA and L-MOSA for a problem with 60 jobs have been shown. There are some solutions in MOSA that are dominated by the solutions in L-MOSA.

Both algorithms have been coded in Matlab 7 language and the instances were tested on a PC, CPU 2 GB RAM and processor 3 GHz.

6 Conclusion

In this paper we proposed a bi-objective simulated annealing algorithm based on Lorenz-dominance properties (L-MOSA) for a problem of scheduling in which rejection of jobs was allowable.

By implementing Lorenz-dominance properties we reduced the number of solutions to facilitate evaluation of choices for decision maker while in some cases, we found even better solutions comparing to the estimated solutions found by MOSA when Pareto-dominance properties were applied.

Considering other dominance concepts, solving this problem by other metaheuristic algorithms like L-NSGA2, particle swarms or ants colony can be an interesting research area for future research. Considering other objective functions in different machine environment is also a potential direction for more research.

References

1. Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., Stougie, L.: Multiprocessor scheduling with rejection. SIAM Journal on Discrete Mathematics 13, 64–78 (2000)
2. Seiden, S.: Preemptive multiprocessor scheduling with rejection. Theoretical Computer Science 262, 437–458 (2001)
3. Hoogeveen, H., Skutella, M., Woeginger, G.J.: Preemptive scheduling with rejection. Mathematical Programming 94, 361–374 (2003)
4. Lu, L., Zhang, L., Yuan, J.: The unbounded parallel batch machine scheduling with release dates and rejection to minimize makespan. Theoretical Computer Science 396, 283–289 (2008)
5. Zhang, L., Lu, L., Yuan, J.: Single machine scheduling with release dates and rejection. European Journal of Operational Research 198, 975–978 (2009)
6. Cheng, Y., Sun, S.: Scheduling linear deteriorating jobs with rejection on a single machine. European Journal of Operational Research 194, 18–27 (2009)
7. Lu, L., Cheng, T.C.E., Yuan, J., Zhang, L.: Bounded single-machine parallel-batch scheduling with release dates and rejection. Computers and Operations Research 36, 2748–2751 (2009)

8. Cao, Z., Yang, X.: A PTAS for parallel batch scheduling with rejection and dynamic job arrivals. Theoretical Computer Science 410, 2732–2745 (2009)
9. Dósa, G., He, Y.: Scheduling with machine cost and rejection. Journal of Combinatorial Optimization 12, 337–350 (2006)
10. Nagy-György, J., Imreh, Cs.: Online scheduling with machine cost and rejection. Discrete Applied Mathematics 155, 2546–2554 (2007)
11. Bansal, N., Blum, A., Chawla, S., Dhamdhere, K.: Scheduling for flow-time with admission control. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 43–54. Springer, Heidelberg (2003)
12. Engels, D.W., Karger, D.R., Kolliopoulos, S.G., Sengupta, S., Uma, R.N., Wein, J.: Techniques for scheduling with rejection. Journal of Algorithms 49, 175–191 (2003)
13. Moghaddam, A., Yalaoui, F., Amodeo, L., Karimi, B., Jolai, F.: Developing a Technique for a Bi-objective Scheduling Model with Rejection by Simulated Annealing. In: MOSIM 2010, Hammamet, Tunisia (2010)
14. Jolai, F., Sangari, M.S., Babaie, M.: Pareto simulated annealing and colonial competitive algorithm to solve an offline scheduling problem with rejection. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, doi: 10.1243/09544054JEM1746
15. Moghaddam, A., Yalaoui, F., Amodeo, L.: Single Machine Scheduling with Rejection: Minimizing total Weighted Completion Time and Rejection Cost. In: EURO XXIV, Lisbon, Portugal (2010)
16. Ulungu, E.L., Teghem, J.: The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. Foundation of Computing and Decision Sciences 20(2), 149–165 (1995)
17. Moghaddam, A., Amodeo, L., Yalaoui, F.: Simulated annealing for a bi-objective scheduling problem with rejection: comparative study. In: META 2010, Djerba, Tunisia (2010)
18. Dugardin, F., Yalaoui, F., Amodeo, L.: New multi-objective method to solve reentrant hybrid flow shop scheduling problem. European Journal of Operational Research 203, 22–31 (2010)
19. Kostreva, M.M., Ogryczak, W., Wierzbicki, A.: Equitable aggregation and multiple criteria analysis. European Journal of Operational Research 158, 362–377 (2004)
20. Golden, B., Perny, P.: Infinite order Lorenz dominance for fair multiagent optimization. In: van der Hoek, W., Kaminka, G.A., Lespérance, Y., Luck, M., Sen, S.: (eds.) Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, pp. 383–390 (2010)
21. Perny, P., Spanjaard, O., Storme, L.-X.: A decision-theoretic approach to robust optimization in multivalued graphs. Annals of Operations Research 147, 317–341 (2006)
22. Moghaddam, A., Amodeo, L., Yalaoui, F.: Single Machine Scheduling with Rejection: Minimizing total Weighted Completion Time and Rejection Cost. Journal of Mathematical Modelling and Algorithms (to be submitted)
23. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. Mathematical Programming 10, 146–175 (1976)
24. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
25. Mansouri, S.A.: A simulated annealing approach to a bi-criteria sequencing problem in a two-stage supply chain. Computers and Industrial Engineering, 1–15 (2006)

26. Varadharajan, T.K., Rajendran, C.: A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. European Journal of Operational Research 167, 772–795 (2005)
27. Ulungu, E.L., Teghem, J., Fortemps, P.H., Tuyttens, D.: MOSA method: A tool for solving multiobjective combinatorial optimization problems. Journal of Multi-Criteria Decision Analysis 8, 221–236 (1999)
28. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 7, 117–132 (2003)
29. Riise, A.: Comparing genetic algorithms and tabu search for multi-objective optimization. In: IFORS, Proceedings of Abstract Conference, p. 29 (2002)

GRACE: A Generational Randomized ACO for the Multi-objective Shortest Path Problem

Leonardo C.T. Bezerra[1,*], Elizabeth F.G. Goldbarg[1,**], Luciana S. Buriol[2], and Marco C. Goldbarg[1,***]

[1] Universidade Federal do Rio Grande do Norte, Natal, RN, Brazil
leo.tbezerra@gmail.com, beth@dimap.ufrn.br, gold@dimap.ufrn.br
[2] Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brazil
buriol@inf.ufrgs.br

Abstract. The Multi-objective Shortest Path Problem (MSP) is a widely studied NP-Hard problem. A few exact algorithms were already proposed to solve this problem, however none is able to solve large instances with three or more objectives. Recently, some metaheuristics have been proposed for the MSP, but little can be said about their efficiency regarding each other, since no comparisons among them are presented in the literature. In this paper an Ant Colony Optimization (ACO) algorithm, called GRACE, is proposed for the MSP. The proposed approach is compared to the well-known evolutionary algorithm NSGA-II. Furthermore, GRACE is compared to another ACO algorithm proposed previously for the MSP. Results of a computational experiment with eighteen instances, with three objectives each, show that the proposed approach is able to produce high quality results for the tested instances.

Keywords: Shortest Path Problem, Multi-objective, Ant Colony Optimization.

1 Introduction

The Shortest Path Problem is a classical problem from graph theory and combinatorial optimization areas. Among several different real-world applications, routing scenarios are of particular interest. On the Internet, for example, determining the best route for sending a package following path-based protocols is an important step for ensuring routing efficiency. On navigation systems, which have become very common even on popular vehicles, shortest paths help on both planning and optimization of the resources. On emergency situations, retrieving minimal routes is critical to the rescue and survival of citizens.

Among the several shortest path problems (all pairs, point-to-point, k-shortest path, among others), in this study we refer to the Point-to-Point Shortest Path Problem simply as the Shortest Path Problem. Although much research has been

[*] Leonardo C.T. Bezerra is supported by CNPq under project no. 579226/2008-5.

[**] Elizabeth F.G. Goldbarg is partially supported by CNPq.

[***] Marco C. Goldbarg is partially supported by CNPq.

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 535–549, 2011.

done on this problem [7,4], usually real-world situations are more accurately represented via multiple criteria. Having more than one objective function to be optimized, multi-objective problems have a set of *non-dominated solutions*, instead of a single optimal solution. For a k-objective minimization problem (without loss of generality) we denote $s_1 \prec s_2$ (solution s_1 *dominates* solution s_2) iff $\forall i = 1, ..., k,\ s_1^i \leq s_2^i$, and $\exists\, i,\ s_1^i < s_2^i$.

The non-dominated solutions are called *efficient solutions*, and the *Pareto optimal set* (or *Pareto set*) contains all such solutions. Each solution can be mapped to the *objective space* through its objective vector value. The set of objective vector values from the Pareto set is called *Pareto front.* Retrieving Pareto sets is the goal of multi-objective algorithms. Depending on the size of the instance, the nature of the problem, and the amount of objectives considered, this task might be unfeasible either by memory or time limitations. MSP, for instance, is an NP-Hard problem [30]. Thus, its Pareto set is expected to grow exponentially.

Several exact algorithms have been proposed for the MSP, and the most efficient relies on a two-phase strategy [28]. On the first phase, the algorithm searches for *supported efficient solutions*, which are solutions that can be retrieved by solving single objective problems obtained with weighted sums of the objectives. These solutions narrow the search space for the second phase, on which *non-supported efficient solutions* are searched (non-supported solutions cannot be retrieved via scalarizations). As far as the authors' knowledge concerns, no two-phase exact algorithms have been proposed for the MSP with more than two objectives.

Some metaheuristics have recently been proposed for solving the MSP using genetic algorithms and ant colony optimization. Their goal is to generate an *approximation set*, either containing suboptimal solutions, or part of the actual Pareto set, or even both. Some of the proposed approaches deal with the Bi-objective Shortest Path problem (BSP), but most focus on three or more objectives, and none uses two-phase strategies. Although several approaches have been proposed, solid multi-objective performance assessment has not been done to evaluate and compare them.

In this paper, a two-phase generational randomized Ant Colony Algorithm named *GRACE*, is proposed for the MSP. In order to find supported efficient solutions, a search strategy called *Logos* is proposed, which divides scalarization vectors in intervals, resembling divide-and-conquer techniques. The two- and three-objective versions of this strategy are detailed. GRACE is compared to two multi-objective algorithms: an ACO published by Häckel *et al.* for the MSP [13], and NSGA-II, a well-known multi-objective evolutionary algorithm proposed by Deb *et al.* [6] (an available implementation [25] was adapted for this problem).

The paper is organized as follows. Section 2 presents the MSP formulation and a literature review on algorithms for solving this problem. In Section 3, the Ant Colony Optimization approach is described, and several ACOs proposed for different multi-objective problems, including the MSP, are revised. GRACE and

Logos are presented in Section 4. The methodology used for the experimentation conducted in this work is described in Section 5, and results are discussed in Section 6. Finally, conclusions and future work are presented in Section 7.

2 The Multi-objective Shortest Path Problem

The Multi-objective Shortest Path problem studied in this paper is a generalization of the classical point-to-point shortest path problem, and is presented by Raith and Ehrgott using a network flow formulation for two objectives [28]. In this paper, we expand this formulation to deal with any number of objectives (assuming positive edge weights):

$$\min \quad z(x) = \begin{cases} z_1(x) = \sum\limits_{(i,j)\in A} c^1_{ij} x_{ij} \\ \quad\quad\quad ... \\ z_k(x) = \sum\limits_{(i,j)\in A} c^k_{ij} x_{ij} \end{cases} \tag{1}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = \begin{cases} 1 & \text{if } i = s, \\ 0 & \text{if } i \neq s, t \\ \text{-1} & \text{if } i = t, \end{cases} \tag{2}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \ . \tag{3}$$

where s and t are, respectively, source and terminal nodes, c is a k-dimensional cost matrix for each edge (i, j), and z is the objective vector composed by k objective functions.

Several exact algorithms have been proposed for the MSP [11]. Skriver [32] proposed a classification for the particular case of two objectives (BSP), and grouped the algorithms as *node labelling* or *path/tree* algorithms. Among the node labelling algorithms, the existing methods were subdivided into two classes: *label setting* and *label correcting*. As for the path/tree family, the author identifies a *two-phase method* and a *k-th shortest path algorithm*. Recently, Raith and Ehrgott [28] compared the algorithms of the first three families using three sets of bi-criteria instances. Their experiment shows that label setting and correcting algorithms present good performance on smaller instances, whereas for large ones their times increase significantly. However, the dichotomic two-phase algorithm (different from the original two-phase algorithm [23]) is efficient even on large instances for the bi-objective case ($k = 2$).

Mooney and Winstansley [22] proposed a multi-objective evolutionary algorithm that has regular and elite populations, and generates the initial individuals through a random walking mechanism [5]. The authors use path encoding (representation by vertices), one-point crossover, binary tournament and a path mutation operator that substitutes the gene of a random locus by a random node. Both crossover and mutation operators only succeed in case the path feasibility is maintained. The experiments showed that the random walking strategy enabled the algorithm to cover a set of nodes and edges considered to be satisfactory by the authors. It also showed that, for three and four objectives, with instances ranging from 100 to 3500 nodes, their algorithm was able to approximate a reference set created by three algorithms: Dijkstra, a k-th shortest path and many executions of their own algorithm. Finally, results show the EA was able to find extreme supported solutions faster than Dijkstra on real-world networks.

He *et al.* [14] also proposed an elitist evolutionary algorithm with two populations, but initial individuals are generated using depth first search mechanism and are ranked according to dominance ranking and niche count. They also use variable length chromosomes (path encoding using vertices), and an one-point crossover with binary tournament: a random cell is chosen from one parent, and in case the same node is also present on the other parent, the crossover is performed (a repair function eliminates possible loops). The mutation operator reconstructs the chromosome from a random locus through depth first search mechanism. An illustrative example showed the efficiency of the proposed MOEA on a 50 nodes three-objective instance.

Pangilinan and Janseens [27] tested SPEA2 [38] for the MSP using the implementation available on PISA framework [1]. They also used path encoding, but generated the initial population randomly. The one-point crossover and the mutation operator are identical to [14], but mutation reconstructs individuals randomly. Using three objectives instances, the approximation set presented good diversity on two of the objectives, but the authors were unable to determine the optimality of the solutions, since the actual Pareto sets were unavailable. Computational results showed that their MOEA is slower than [20].

Lin and Gen [18] presented a multi-objective evolutionary algorithm for the BSP. Their adaptive weight algorithm uses priority-based encoding (fixed chromosome sizes), roulette selection, weight-mapping crossover, a random mutation operator and also an immigration operator [21]. No information is given on how the initial population is generated. The authors also use two fuzzy logic controllers in order to auto-tune the parameters of their algorithm. They compare their MOEA against implementations of NSGA-II [6], SPEA [36] and rwGA [16] that use the same encoding and operators, analyzing diversity, cardinality, ratio of non-dominated solutions and computational time. Regarding these indicators, results show that the priority-based encoding and the auto-tuning are good strategies, and that their algorithm outperform the others.

As described above, several evolutionary algorithms have been proposed for the MSP over the last few years. However no comparison has been presented to evaluate whether one algorithm outperforms the others. In the next section, we revise the Ant Colony Optimization metaheuristic, and the literature on multi-objective ACOs, including the MSP.

3 Ant Colony Optimization

Ant Colony Optimization is a bio-inspired metaheuristic that uses the concept of *swarm intelligence* and *stigmergy*. It was originally proposed by Dorigo [10] as the *Ant System* (AS), and lately improved into *Ant Colony System* (ACS) [8].

The original AS consists of *agents* (ants) that are able to *evaluate their current state* and choose the next through a set of possible *transitions*. Each transition is weighted with *pheromone* and *heuristic* information (random proportional rule). The *constructive procedure* allows these agents to generate *viable solutions* for a combinatorial optimization problem. The ants are also able to *deposit* pheromone over the tracks they visit, which naturally *evaporates* with time. Two major changes were proposed to improve AS into ACS. First, when choosing the next state, a random value is tested against a threshold. Depending on that result, the choice might be for the best transition or using the original random proportional rule. Second, ants are allowed to update pheromone information as soon as they finish their constructive procedure. In AS, all ants perform pheromone update altogether when all of them finish building their solutions.

Dorigo and Socha [9] list some ACOs proposed for multi-objective problems. Iredi *et al.* [15] affirm some of these ACOs either consider some objectives to be more important than others, or deal with problems in which each objective can be treated separatedly. In this work, we limit ourselves to detailing the ACOs designed for problems with equally important objectives. In 2000, Iredi *et al.* [15] proposed a multi-colony ant system for a scheduling problem. In their work, multiple colonies are used to search different regions of the Pareto front. Each ant has a scalarization vector which is used for weighting heuristic information in the random proportional rule. Two pheromone matrices are used, one for each objective, and are updated as in ACS. Only ants that find new non-dominated solutions regarding a global archive are allowed to update the pheromone matrices. Lopéz-Ibanéz *et al.* [19] published a comparative study of different strategies for multi-objective ACO using local search strategies on the Bi-objective Quadratic Assignment Problem. The authors reported that results differ depending on the correlation of the objectives, but the usage of local search procedures improved the overall algorithm performance.

For the MSP, as far as the authors are aware of, only two ACOs have been proposed. Häckel *et al.* [13] proposed a version of the algorithm presented in [15] for the MSP. The algorithm proposed by Häckel et al. [13], referred to as HACO in this paper, do not allow two ants searching on the same region of the Pareto front (called overlapping zones [15]). The heuristic information comes from a Dynamic Programming algorithm called *Look-Ahead Heuristic* (LAH). As for the pheromone matrices, the authors present equations as if multiple matrices were used, but state that only one matrix is used. The experiments conducted in their work show that the usage of LAH improves overall results, and that their ACO obtains diverse solutions in comparison to a dynamic multi-objective algorithm not referenced by the authors on three objective instances[1].

Ghoseiri and Nadjari [12] proposed an ACO for the BSP and compared it to a label correcting algorithm, also not referenced. A single colony and two pheromone matrices (one for each objective) were used. As for the heuristic information, two sources were used: normalized edge weights and number of nodes

[1] The authors do not explicit instances size, but state the edges plus vertices amount range from 527 to 1408.

to the destiny node. Scalarizations are used to compute both pheromone and heuristic information. This algorithm resembles ACS, except for the fact that local pheromone update is performed after each transition. The experimentation conducted by the authors used two objective NetMaker [31] instances. Results showed their algorithm was able to generate a diverse approximation set, but the Pareto fronts graphically presented no efficient solutions. Their ACO was computationally faster than the label correcting method. There is also no comparison between the proposed multiobjective ACOs for MSP and other metaheuristics.

4 GRACE: A Two-Phase ACO for the MSP

In this section, we present GRACE (Generational Randomized Ant Colony Enhancement) and the two-phase strategy (including the Logos procedures). The initial review on exact algorithms pointed that two-phase algorithms are efficient for the BSP, which led us to work on scenarios with three or more objectives. Since no two-phase strategy has been proposed for the shortest path problem with more than two objectives, we review the literature of other problems, and finally propose our own search strategy: a two-phase ACO for the MSP.

As seen in Section 3, many strategies have been proposed for multi-objectives ACO, but as reported by [19] the efficiency of the strategies depends much on the problem and on the instances. Hence, we chose a particular configuration and tested it against two algorithms from the literature. We chose one algorithm to represent each class of metaheuristics studied for the MSP. Since none of the MOEAs proposed had available implementations[2], we decided to use NSGA-II [6], a well-known MOEA with an available implementation for experimentation [25]. As for the ACOs, we implemented [13] according to the authors description.

4.1 Phase I: The Quest for Supported Efficient Solutions

In our review of search strategies for supported efficient solutions, we outline two methods found in the literature of MO problems with $k > 2$. Murata *et al.* [26] proposed a uniform distribution of scalarization vectors over the Pareto optimal surface. Their approach generates sequential weights separated by an ϵ gap. For large values of ϵ the algorithm is faster, but possibly miss many solutions, whereas for small values a larger set will be retrieved at the expense of higher computational times. Rocha [29] proposed a stochastic version of this algorithm, as not all of the weights are generated. Starting at the canonical scalarization vectors, the main objective scalarization value is decremented by ϵ and another random objective scalarization value is incremented by ϵ. The author found good results for the Multi-objective Quadratic Assignment Problem, but for large instances some parts of the Pareto front is not searched, resulting in loss of solutions.

[2] Each author was properly contacted through the emails listed on their papers.

In this work, we adapted the general idea present in the *geometric* [33] and the *dichotomic* [28] bi-criteria strategies in order to apply them to instances of three objectives. Both strategies start at the extreme supported efficient solutions s_1 and s_2, calculate an intermediary scalarization vector through evaluation functions, and test the existence of a new supported efficient solution s'. If the non-dominance of s' is confirmed, the search recursively proceeds for (s_1, s') and (s', s_2). Following the general idea, 2-Logos (Logarithmic 2-Objective Space Search) iteratively divides the region of the Pareto front, resembling a logarithmic function. Given two different non-dominated solutions s_i and s_f, found on (x_i, y_i) and (x_f, y_f) of the Pareto front respectively, the solution s_{mid}, from $(x_{midpoint}, y_{midpoint})$, is tested (lines 1-2). If it is a different non-dominated solution, the procedure is recursively called for (s_i, s_{mid}) and (s_{mid}, s_f) (lines 3-4). Otherwise, the recursion stops.

```
procedure 2-Logos
1:  {Require Pareto set S, solution si, solution sf};
2:  s_mid = Dijkstra(midpoint(si.weights, sf.weights));
3:  if (newSolution(S, s_mid)) then
4:      2-Logos(S, si, s_mid);
5:      2-Logos(S, s_mid, sf);
6:  end if
end
```

For three-objective instances, 3-Logos (Logarithmic 3-Objective Space Search) is proposed, and its pseudocode is presented below. 3-Logos initially finds extreme supported efficient solutions using Dijkstra algorithm. These three solutions are considered vertices of a triangle where each edge is formed by the line that connects each pair of these solutions (algorithm input). Each edge is scanned by the 2-Logos procedure (line 1). After completion of 2-Logos executions on the three edges (line 2), the centroid of the triangle is calculated and, in case the obtained solution is new in the set S of non dominated solutions (Pareto front), 3-Logos is recursively called for these three new subtriangles (lines 4-6). Both 2- and 3-Logos are illustrated in Figure 1.

```
procedure 3-Logos
1: {Require Pareto set S, solution sa, solution sb, solution sc};
2: S = addSolutions(S, 2-Logos(sa, sb), 2-Logos(sa, sc),
    2-Logos(sb, sc));
3: s_ctr = Dijkstra(centroid(sa.weights, sb.weights, sc.weights));
4: if (newSolution(S, s_ctr)) then
5:    3-Logos(S, sa, sb, s_ctr);
6:    3-Logos(S, sa, s_ctr, sc);
7:    3-Logos(S, s_ctr, sb, sc);
8: end if
end
```

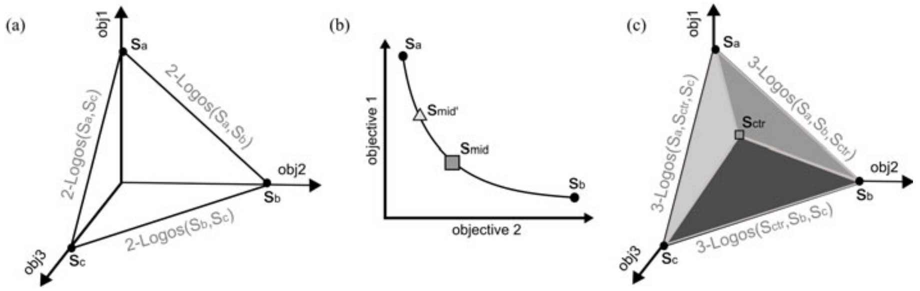


Fig. 1. Example of one iteration of 3-Logos

4.2 Phase II: The Ant Colony

For the second phase of our ACO, we chose to create a single colony algorithm, in which iterations are called generations. Every ant is going to search the same region of the Pareto front using a scalarization vector randomly created for each generation. The purpose of this strategy is to allow a significant amount of ants to explore the same region. A single pheromone matrix is used in our ACO, and is warmed up at the end of Phase I. For every edge present in the efficient supported paths, a pheromone deposit $\tau_{deposit}$ is added, as many times as the edge appears in the set. An ant is only allowed to update the pheromone matrix when it finishes building its solution, and only if that solution is non-dominated regarding the non-limited global archive. For the heuristic information, we improved the idea used in [13] by using Dijkstra algorithm (we are assuming non-negative weight inputs). The heuristic information represents the distance, under a specific scalarization vector, from each node to the destination. The stopping criterion is a number R of generations with no generation of a new non-dominated solution regarding the global archive.

The pseudocode of GRACE is presented in the following page. Initially, extreme supported efficient solutions are found[3] (lines 1-2) and 3-Logos is called (line 3). The pheromone matrix is warmed up using the supported efficient solutions (line 4), and the generational cycle starts (line 5). First, a common scalarization vector is randomly created (line 6), and the heuristic information is calculated (line 7). Then, ants build their solutions (lines 8-14) and, if a new non-dominated solution regarding the global archive is found, a pheromone update is performed (lines 10-12). In case a new non-dominated solution was found over the last cycle, the algorithm is allowed to start a new one (line 15).

```
Algorithm GRACE
1: {Require graph G, vertex s, vertex t};
2: extreme_solutions = Dijkstra(canonicalVectors());
3: supported = addSolutions(supported, extreme_solutions);
```

[3] In this work, we refer to the solutions found for scalarizations using canonical vectors as extreme supported efficient solutions.

```
3-Logos(supported, extreme_solutions));
pheromoneM = pheromoneWarmup(supported);
for (i = 0; i < R; i++)
   l = randomScalarizationVector();
   heuristicM = heuristicInfo(Dijkstra(reverse(G), t, l));
   for (j = 0; j < n_ants; ++j)
      s = buildSolution(pheromoneM, heuristicM, s, t, l);
      if (newSolution(unsupported, s)) then
         unsupported = addSolution(unsupported, s);
         pheromoneUpdate(s);
      end if
    end for
    if ((i == R) and (newSolution(cycle))) then i = 0;
 end for
end
```

5 Methodology

To compare GRACE against NSGA-II and the ACO of Häckel *et al.* [13], we adapted the existing NSGA-II implementation [25] for the MSP. We decided to use path encoding, which is also the choice made by most MOEAs proposed for this problem [14,22,27]. We also used crossover [14] and mutation [27] operators found in the MSP literature. The repair function of the crossover operator was redefined: the chromosome is scanned from the first to the last locus, and every time a duplicated vertex is found, all vertices between them are removed. We also used a specific data structure to store outgoing vertices and speed up the algorithm [2]. The algorithm of Häckel *et al.* [13], named HACO in this paper, was implemented following the directions given by the authors.

Since multi-objective optimizers produce approximation sets instead of single solutions, a specific methodology must be used for their evaluation. The guidelines for performance assessment utilized in this work are reviewed by Knowles *et al.* [17]. We use dominance ranking with the binary epsilon indicator [37] and, if necessary, the unary quality indicators I_H^1 [36] and $I_{\epsilon+}^1$ [37]. Reference sets are generated by merging all the approximation sets generated by the optimizers being compared, and removing the dominated solutions.

Tests were executed on an Intel Core 2 Duo @ 2.2GHz, with 1Gb of RAM and a Linux Ubuntu 9.04 distribution. We used a set of 18 instances, generated by Santos [24], from two distinct classes: *grid*, which represents a square grid, and *complete*, which contains complete graphs. For our experiment, we used fixed time limits per instance as the stopping criterium. In most cases, these limits were set according to instance size and class. Some details of the instances are presented in Table 1, where column # shows the instance identification, *Type* shows the type of graph (complete, grid), $|N|$, $|A|$ and k are respectively the number of vertices, edges and objectives of each instance, and finally *t(s)* is the time limit in seconds for the executions of the algorithms.

Table 1. List of instances. Only the first three objectives were used on *large* instances.

#	*Type*	$\|N\|$	$\|A\|$	k	$t(s)$	*Seed*
1	CompleteN-small	25	600	3	5	45
2	CompleteN-small	50	2450	3	10	12
3	CompleteN-small	100	9900	3	12	13
4	CompleteN-medium	40	780	3	5	18
5	CompleteN-medium	120	14280	3	10	14
6	CompleteN-medium	200	39800	3	15	21
7	CompleteN-large	100	9900	6	8	1
8	CompleteN-large	150	22350	6	40	10
9	CompleteN-large	200	39800	6	40	1
10	GridN-small	64	224	3	7	14
11	GridN-small	144	528	3	36	1
12	GridN-small	256	960	3	81	26
13	GridN-medium	484	1848	3	100	1
14	GridN-medium	961	3720	3	100	40
15	GridN-medium	1225	4760	3	100	2
16	GridN-large	121	440	6	60	41
17	GridN-large	484	1848	6	100	42
18	GridN-large	900	3480	6	100	43

In order to fine-tune the NSGA-II, we tested several values for population size, number of generations, crossover and mutation rates. Since we used fixed time limits, we initially defined a large number of generations to find out the maximum population size the algorithm could stand: 200 generations for complete instances and 100 for grid instances. Complete instances are divided into two groups: instances 1 to 7 and instances 8 and 9, since results showed time limits were critical to population size. A linear function $pop(x) = a.t(x) + b$ was used to set the values of population size for grid instances. Table 2 shows the values used to each test configuration. Crossover rates 0.7, 0.8 and 0.9 were tested, as well as values 0.01, 0.005 and 0.001 for mutation rates [6]. Dominance ranking with the binary epsilon indicator [37] pointed population size of 2500 & 3000 to be statistically significantly better for complete instances. As for grid graphs, neither the dominance ranking nor the unary quality indicators I_H^1 [36] and $I_{\epsilon+}^1$ [37] were conclusive[4]. For crossover rates, no statistical significant differences was observed for complete instances, and for grid instances results were inconclusive. For mutation rates, no statistical significant differences was observed.

Table 2. Parameter values tested for NSGA-II

#config	Complete 1-7	Complete 8-9	Grid a	Grid b
1	1000	1500	20	600
2	1500	2000	20	900
3	2500	3000	25	800

[4] When a configuration proved to be better for only few instances, we considered results to be inconclusive.

For HACO, we used the parameter setting presented by the authors in their paper. The only parameter not informed by the authors was the penalty parameter ξ. We tested several values ($\xi \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$), but no statistical significant difference was found.

The final parameter settings were: (i) NSGA-II: *pop*=2500/3000, *a*=25, *b*=800, *cross*=0.9, *mut*=0.005; (ii) Hackel: $\chi_{colonies} = 3, \chi_{ants} = 4$, $\alpha = 0.5$, $\beta = 3$, $\tau_0 = 0.05$, $\rho = 0.1$, $q_0 = 0.5$, $\xi = 0.8$, and; (iii) GRACE: $n_{ants} = 300$, $\alpha = 0.6$, $\beta = 0.6$, $\tau_0 = 1$, $\tau_{deposit} = 10$, $R = 5$.

6 Results and Discussion

For comparing the results from the different algorithms, we perform a serie of dominance ranking [17] and statistical tests [3][35] over the results. As we can see in Table 3, the p-values of Kruskal-Wallis test on dominance ranking results indicate that there is a statistical significant difference between the dominance rankings of each optimizer. We then proceeded to the pairwise comparison using Wilcoxon test.

Table 3. *p*-values of Kruskal-Wallis test on dominance rankings from all optimizers

#inst	*p*-value	#inst	*p*-value
1	**1.27884e-14**	10	**8.9668e-16**
2	**8.78175e-13**	11	**5.87126e-12**
3	**2.80243e-15**	12	**4.50099e-15**
4	**1.29004e-10**	13	**1.35497e-14**
5	**1.90311e-09**	14	**1.81209e-15**
6	**2.14250e-13**	15	**4.86612e-16**
7	**1.66174e-10**	16	**2.24666e-11**
8	**0.00508**	17	**1.22460e-14**
9	**3.09800e-13**	18	**3.97783e-16**

Table 4 shows *p*-values from both Wilcoxon test (the two-tailed and the one-tailed with "less" as hypothesis) for the comparison between GRACE and NSGA-II. p-values on the "less" column lower than 0.05 indicate that the dominance ranking of GRACE was better, whereas p-values greater than 0.95 indicate the dominance ranking of NSGA-II was better. For all instances (both grid and complete), results show GRACE generates statistically significantly better approximation sets than NSGA-II.

Next, we compare GRACE with HACO. Results on Table 5 show statistical significant difference for 13 instances, and in all of them the dominance rankings of GRACE are attested statistically better than the ones found by HACO.

Table 4. p-values of Wilcoxon test on dominance rankings for GRACE and NSGA-II

#inst	two-tailed	less	#inst	two-tailed	less
1	**1.09789e-09**	**5.49383e-10**	10	**8.79965e-11**	**4.39982e-11**
2	**3.64852e-10**	**1.82426e-10**	11	**6.28895e-08**	**3.14447e-08**
3	**1.86299e-10**	**9.31494e-11**	12	**0.01044**	**0.00522**
4	**3.98723e-09**	**1.99361e-09**	13	**0.00245**	**0.00012**
5	**1.65375e-07**	**8.26976e-08**	14	**4.20083e-09**	**2.100411e-09**
6	**3.08372e-10**	**1.54186e-10**	15	**4.87374e-11**	**2.43687e-11**
7	**1.31777e-08**	**6.58886e-09**	16	**4.27032e-11**	**2.13516e-11**
8	**0.02074**	**0.01037**	17	**0.00054**	**0.00027**
9	**1.50692e-10**	**7.53461e-11**	18	**9.31587e-11**	**4.365794e-11**

On the remaining 5 instances, unary quality indicators are necessary (Table 6). For the $I^1_{\epsilon+}$ indicator, GRACE approximation sets are considered better, but the I^1_H considers the opposite. This result means that the sets are incomparable [17]. Using the proportions comparison test proposed by Taillard *et al.* [34], p-values for the hypothesis that GRACE generates better approximation sets than HACO are: for complete set, p-value=0.01; for grids, p-value<0.01.

Table 5. p-values of Wilcoxon test on dominance rankings for GRACE and HACO. NaN means rankings were equal.

#inst	two-tailed	less	#inst	two-tailed	less
1	**1.03627e-09**	**5.18137e-10**	10	**3.31826e-11**	**1.65913e-11**
2	**0.03124**	**0.01562**	11	**1.13077e-09**	**5.65385e-10**
3	0.16151	0.92746	12	**2.77473e-12**	**1.38624e-12**
4	0.26094	0.13047	13	**2.77473e-12**	**1.38624e-12**
5	0.18886	0.90887	14	**4.43246e-12**	**2.21629e-12**
6	0.49546	0.24773	15	**1.34262e-11**	**6.71311e-12**
7	**0.01460**	**0.00730**	16	**3.42463e-08**	**1.71232e-08**
8	NaN	1	17	**2.77247e-12**	**1.38624e-12**
9	**1.51267e-05**	**7.56335e-06**	18	**4.43246e-12**	**2.21623e-12**

Results for the last comparison, between NSGA-II and HACO, are listed in Table 7. For complete instances, HACO generates better approximation sets for 8 out of 9 test-cases, whereas for grid instances NSGA-II generates better sets for all 9 instances.

Table 6. p-values of Wilcoxon test on quality indicators for GRACE and HACO

#inst	$I^1_{\epsilon+}$ two-tailed	$I^1_{\epsilon+}$ less	I^1_H two-tailed	I^1_H less
3	**5.34323e-08**	**2.67162e-08**	**1.58215e-14**	**1**
4	**2.71058e-10**	**1.35530e-10**	**1.32748e-09**	**1**
5	**9.79730e-11**	**4.89965e-11**	**1.21274e-09**	**1**
6	**6.59836e-10**	**3.29913e-10**	**1.58215e-14**	**1**
8	**7.63654e-08**	**3.81827e-08**	**1.58215e-14**	**1**

Table 7. p-values of Wilcoxon test on dominance rankings: NSGA-II [6], Hackel [13]

#inst	two-tailed	less	#inst	two-tailed	less
1	**0.03232**	**0.01616**	10	**5.72682e-10**	**2.86341e-10**
2	**4.02499e-07**	**0.99998**	11	**0.02284**	**0.01142**
3	**1.11944e-09**	**1**	12	**2.77247e-12**	**1.38624e-12**
4	**0.03177**	**0.98487**	13	**1.35016e-11**	**6.75080e-12**
5	**1.37130e-08**	**1**	14	**8.04777e-11**	**4.02388e-11**
6	**3.05784e-10**	**1**	15	**1.48317e-10**	**7.41586e-11**
7	**3.32396e-08**	**1**	16	**0.00116**	**0.00058**
8	**0.020744**	**0.99061**	17	**4.43246e-12**	**2.21623e-12**
9	**3.48046e-10**	**1**	18	**1.48645e-11**	**7.43227e-12**

7 Conclusions

In this work we reviewed the MSP literature, including exact algorithms, evolutionary algorithms, and ant colony optimization, and highlighted the lack of comparisons among the metaheuristics. We proposed a two-phase ACO, named GRACE, with a supported efficient solutions search called Logos. For the evaluation of GRACE, we conducted a solid performance assessment against NSGA-II, a well-known MOEA and HACO, an ACO published for the MSP. For 18 instances from classes grid and complete, dominance ranking and statistical tests attested that GRACE generates better approximation sets than NSGA-II. In comparison with HACO, GRACE sets are better for 4 out of 9 complete instances, and 9 out of 9 grid. For the remaining complete instances, GRACE produces sets which are incomparable with HACO's according to two quality unary indicators, I^1_H and $I^1_{\epsilon+}$). When comparing NSGA-II and HACO, the former generates statistically better sets than the latter for 1 out of 9 complete instances, and for all 9 grid instances. For the remaining complete instances, HACO was attested to generate better sets. As future work possibilities, a comparison including a MOEA proposed specifically for MSP could be used, as well as a larger instance database. Another promising work is the comparison of different multi-objective ACO strategies on the MSP, to test which configurations are more efficient.

References

1. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A platform and programming language independent interface for search algorithms. In: Fonseca, C., Fleming, P., Zitzler, E., Thiele, L., Deb, K. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
2. Buriol, L.S., Resende, M.G.C., Thorup, M.: Speeding up dynamic shortest path algorithms. INF. J. on Comp. 20(2), 191–204 (2008)
3. Conover, W.J.: Practical non-parametric statistics, 3rd edn. John Wiley and Sons, New York (1999)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press and McGraw-Hill (2001)
5. Costelloe, D., Mooney, P., Winstansley, A.: From random walks to pareto optimal paths. In: Proceedings of the 4th Irish Artificial Intelligence and Cognitive Science Conference, pp. 523–530 (2001)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. on Evol. Comp. 6(2), 182–197 (2002)
7. Dijkstra, E.W.: A note on two problems in connexion with graphs. Num. Math. (1959)
8. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evol. Comp. 1(1), 53–66 (1997)
9. Dorigo, M., Socha, K.: An introduction to ant colony optimization. Tech. Rep. TR/IRIDIA/2006-010, Institut de Recherches Interdisciplinaires et de Développements em Intelligence Artificialle, Université Libre de Bruxelles (2006)
10. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Dipartamento de Elettronica, Politecnico de Milano (1992)
11. Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spek. 22(4), 425–460 (2000)
12. Ghoseiri, K., Nadjari, B.: An ant colony optimization algorithm for the bi-objective shortest path problem. App. Soft Comp. 10(4), 1237–1246 (2010)
13. Häckel, S., Fischer, M., Zechel, D., Teich, T.: A multi-objective ant colony approach for pareto-optimization using dynamic programming. In: GECCO 2008, pp. 33–40. ACM, New York (2008)
14. He, F., Qi, H., Fan, Q.: An evolutionary algorithm for the multi-objective shortest path problem. In: Proceedings of ISKE 2007 (2007)
15. Iredi, S., Merkle, D., Middendorf, M.: Bi-criterion optimization with multi colony ant algorithms. In: Zitzler, E., Thiele, L., Deb, K., Coello, C.A.C., Corne, D. (eds.) EMO 2001. LNCS, vol. 1993, pp. 359–372. Springer, Heidelberg (2001)
16. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans. on Sys. Man and Cyber. Part C: App. and rev. 28(3), 392–403 (1998)
17. Knowles, J.D., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Tech. Rep. TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich (2006)
18. Lin, L., Gen, M.: A bicriteria shortest path routing problems by hybrid genetic algorithm in communication networks. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 4577–4582 (2007)
19. López-Ibanez, M., Paquete, L., Stützle, T.: On the design of ACO for the biobjective quadratic assignment problem. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L., Montada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 214–225. Springer, Heidelberg (2004)

20. Martins, E.Q.V.: On a multicritera shortest path problem. Eur. J. of Op. Res. 16, 236–245 (1984)
21. Michael, C.M., Stewart, C.V., Kelly, R.B.: Reducing the search time of a steady state genetic algorithm using the immigration operator. In: Proceedings of the IEEE Congress on Tools for AI, pp. 500–501 (1991)
22. Mooney, P., Winstansley, A.: An evolutionary algorithm for multicriteria path optimization problems. Int. J. of Geo. Inf. Sci. 20(4), 401–423 (2006)
23. Mote, J., Murphy, I., Olson, D.L.: A parametric approach to solving bicriterion shortest path problems. Eur. J. of Op. Res. 53, 81–92 (1991)
24. MSPP: www.mat.uc.pt/~zeluis/INVESTIG/MSPP/mspp.htm
25. Multi-objective NSGA-II code in C. Revision 1.1 (June 10, 2005) (For Linux only), www.iitk.ac.in/kangal/codes.shtml
26. Murata, T., Ishibuchi, H., Gen, M.: Specification of genetic search directions in cellular multi-objective genetic algorithms. In: Zitzler, E., Thiele, L., Deb, K., Coello, C.A.C., Corne, D. (eds.) EMO 2001. LNCS, vol. 1993, pp. 82–95. Springer, Heidelberg (2001)
27. Pangilinan, J.M.A., Janseens, G.K.: Evolutionary algorithms for the multiobjective shortest path problem. Int. J. of Comp. and Inf. Sci. and Eng. 1 (2007)
28. Raith, A., Ehrgott, M.: A comparison of solution strategies for the biobjective shortest path problem. Comp. and Op. Res. 36(4), 1299–1331 (2009)
29. Rocha, D.A.M.: Busca local para o problema quadrático de alocação multiobjetivo (pqa-mo). Final essay for Computer Science B.Sc. Universidade Federal do Rio Grande do Norte, Natal, RN, Brazil (2006)
30. Serafini, P.: Some considerations about computational complexity for multi objective combinatorial problems. In: Jahn, J., Krabs, W. (eds.) Recent advances and historical development of vector optimization. Lecture Notes in Economics and Mathematical Systems, vol. 294. Springer, Berlin (1986)
31. Skriver, A.J.V., Andersen, K.A.: A label correcting approach for solving bicriterion shortest-path problems. Comp. and Op. Res. 27(6), 507–524 (2000)
32. Skriver, A.J.V.: A classification of bicriterion shortest path (bsp) algorithms. Asia-Pac. J. of Op. Res. 17, 199–212 (2000)
33. Steiner, S., Radzik, T.: Solving the biobjective minimum spanning tree problem using a k-best algorithm. Tech. Rep. TR-03-06, Department of Computer Science, King's College London (2003)
34. Taillard, E., Waelti, P., Zuber, J.: Few statistical tests for proportions comparison. Eur. J. Oper. Res. 185, 1336–1350 (2009)
35. Wilcoxon, F.: Individual comparisons by ranking methods. Bio. bull. 1(6), 80–83 (1945)
36. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto evolutionary algorithm. IEEE Trans. on Evol. Comp. 3(4), 257–271 (1999)
37. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. Tech. Rep. TIK-Report No. 139, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich (2002)
38. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. Tech. Rep. TIK-Report 103, Computer Engineering and Networks Laboratory (TIK), Departament of Electrical Engeneering, Swiss Federal Institute fo Technology (ETH) Zurich (2001)

Modeling Decision-Maker Preferences through Utility Function Level Sets

Luciana R. Pedro[1] and Ricardo H.C. Takahashi[2]

[1] Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
[2] Department of Mathematics
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
taka@mat.ufmg.br

Abstract. In this paper, we present a method based on the multiattribute utility theory to approximate the decision-maker preference function. A feature of the proposed methodology is its ability to represent arbitrary preference functions, including functions in which there are non-linear dependencies among different decision criteria. The preference information extracted from the decision-maker involves ordinal description only, and is structured using a partial ranking procedure. An artificial neural network is constructed to approximate the decision-maker preferences, reproducing the level sets of the underlying utility function. The proposed procedure can be useful when recurrent decisions are to be performed, with the same decision-maker over different sets of alternatives. It is shown here that the inclusion/exclusion of information causes only local rank reversals instead of large scale ones that may occur in several existing methodologies. The proposed method is also robust to relatively large levels of wrong answers of the decision maker.

Keywords: Multicriteria decision analysis, MAUT, artificial neural networks, utility function.

1 Introduction

The decision-aiding methodologies often assume that it is possible to sort the existing alternatives to solve a problem through the decision-maker (DM) preferences. This sorting can be used to identify the preferred alternative within the set of possible solutions or to classify the set elements in categories. Currently, there are two main theoretical tendencies in mathematical modeling based decision making: the decision based on the multiattribute utility theory (MAUT) and the decision based on outranking relations (OR).

The canonical multiattribute utility theory (MAUT) assumes that there exists a function U, denoted utility function, which represents the decision-maker preferences. This function assigns a scalar value to the alternatives, which can be sorted by the simple comparison of the values [8]. The usage of the MAUT-based

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 550–563, 2011.

methods is appropriate for cases in which there is a previous complete knowledge of all necessary information about the problem, which allows well-structured preferences for the DM. Amongst MAUT-based methods, we can cite: Smarts and Smarter [6], Weighted Sum Model [7], Weighted Product Model [12], AHP [14] and ANP [15].

In 1968, Bernard Roy proposed the concept of outranking relations (OR) [13]. The methods based on OR have been developed to deal with situations which cannot be modeled by MAUT-based methods. In general, these methods are characterized by two steps: construction of the outranking relation and exploitation of the results obtained in the previous stage. The usage of OR methods is relevant when the DM does not have, at the initial stage, either a total knowledge of the preferences or a total knowledge of the available alternatives. In general, these methods involve more complex algorithms, but they demand less previous information, because they assume that the DM is constructing the preferences during the application of the method. Among the methods based on outranking relations, we can cite: Promethee [4], Electre [13] and their variations.

In both MAUT and OR current methodologies, the problem setting is stated usually as: given a problem, with its set of possible solutions (the alternatives), establish a rational route to find a satisfactory solution, under the viewpoint of the DM. In both MAUT and OR methodologies, it is frequently assumed that the several decision criteria are essentially independent: each criterion is analyzed separately, and an aggregation of the multiple criteria is performed using some pre-defined aggregation function. It is worthy to notice that an important existing approach for the problem of representing non-linear dependencies between different criteria in decision problems is based on Choquet integrals [1,2]. However, this approach requires that the decision-maker provides information about the criteria, and not about the solutions directly.

In this paper, we deal with a slightly different problem setting: it is assumed here that the DM preference function may be arbitrary, possibly presenting non-linear dependencies among several decision criteria. The specific structure of interaction with the decision-maker assumed here considers a kind of situation in which the DM should evaluate a solution as a whole, instead of weighting the criteria which should be used in order to evaluate a solution. For instance, in the case of automatic image generation, it would be meaningless to ask a DM what is the relative importance of features such as brightness or contrast. A more meaningful query would be formulated as what is the preferred image, considered some given alternatives. We devise several contexts in which such an interaction structure would be useful, for instance in the evaluation of automatic art systems (automatic music composition, automatic picture), in consumer preference modeling, and so forth.

The aim of this paper is to present a methodology for the obtention of such preference structure, in the form of an explicit function that reproduces the preference relations extracted of the DM. This function performs a kind of *regression* on the DM answers about the preferences, delivering new answers to alternatives that have not been presented yet. The assignment of space coordinates to the alternatives provides a geometric structure to the utility function, making possible a regression process. This means that alternatives with similar coordinates in the feature space (the space in which the available alternatives with the corresponding decision parameters are embedded) should have similar preference values, i.e., the utility function should be modeled as a continuous function. As a consequence, a regression of utility function values may be meaningful, and may help to guide a search for the preferred alternative from a set of alternatives, even when none of such alternatives has been considered yet – relying only on the information about other points that belong to the same region of the space. The method proposed here employs some aspects of this geometric structure to find an approximation for the DM preferences. As a regression method, we employ multi-layer perceptron artificial neural networks (ANN) trained with a multiobjective procedure that aims to guarantee the regularity of ANN response [16].

It should be noticed that some former works have already exploited the idea of representing DM preferences using artificial neural networks. The reference [5] has reported good results of application of an ANN in order to extract DM preferences, in a problem setting which is similar to the one assumed here. There are, however, three main differences between the methodologies: (i) The methodology of [5] is oriented toward the determination of the "most preferred solution" in a single decision-making problem. Instead, the methodology presented here is intended to build a map of the DM preferences, useful in successive instances of the same decision-making problem. As a consequence, the methodology proposed here involves the construction of partial orderings of the alternative set in order to support such a mapping, which is not performed in [5]. (ii) Also, in [5] a kind of rough quantitative information concerning the preferences is assumed. At least some approximate values of preference ratios are required. In the methodology presented here, it would be possible to use only answers to yes/no queries – indeed, in the presentation of this communication, only such kind of answers are assumed. (iii) Finally, the architectures of the artificial neural networks employed here and in [5] are quite different. The architecture employed in [5] involves two ANNs, which process the criterion vectors, leading to results whose ratio is calculated, being delivered as the final result. This architecture is suitable for the purpose of discriminating the "most preferred alternative". On the other hand, the architecture employed here involves training a single ANN, which is the most suitable arrangement for delivering a map of the DM preferences.

1.1 Notation and Problem Statement

A multicriteria decision making problem usually involves the following basic elements:

- *set A of alternatives (possible actions or choices).* This set can be discrete or continuous and it is considered the domain of the decision making problem. Each element $a \in A$ corresponds to an available alternative and each feature of a provides a problem dimension.

- *set B of consequences or attributes.* Each alternative a in set A has attributes, which reflect the consequences of its execution for each decision criterion: $b = f(a)$, $b \in B$.
- *decision-maker.* The *value* of each alternative is assigned by a decision-maker, that formally corresponds to a *preference function* $\mathcal{P}$. The best alternative $x^* \in A$ is the one which has attributes $f(x^*) \in B$ that maximize the function $\mathcal{P}(f(x)) = P(x)$ in the set A. It is assumed here that it is not possible to directly measure the values of $P(x)$, for any alternative x. Only the ordinal information, of the form $P(x_i) > P(x_j)$, or $P(x_i) = P(x_j)$, may be extracted from yes/no queries to the DM.

In this paper, we are interested in a class of decision-making problems in which the possible solutions to the problem are directly presented to the DM. The decision-maker will provide answers to queries concerning her/his preferences, leading to the discovery of a model for these preferences. Assuming the same preference function P in all problem instances, this model is able to find the best alternative in any instance, considering the available alternatives in each case. The following elements are involved:

- *set $\mathcal{A}$ of all possible alternatives that may appear in each decision problem instance;*
- *set $A_i \subset \mathcal{A}$ of alternatives that are available in the instance i of the decision problem class;*
- *set B_i of consequences or attributes associated to the alternatives in set A_i, for each instance i;*
- *decision-maker; the* **preference function** $P : \mathcal{A} \mapsto \mathbb{R}$, *which is assumed to be the same for all decision problem instances.*

This paper presents a methodology for the construction of a function which models the preferences of a decision-maker. This methodology is compatible with the assumptions of multiattribute utility theory. Such a function is built from a partial ranking process, which is based on the ordinal information provided by the decision-maker, and is used to quantify the preferences within the domain $\mathcal{A}$. An artificial neural network is used to construct a function that should have level sets which coincide with the ones of the decision-maker utility function. In the specific domain $\mathcal{A}$, this utility function approximation provides a model for the decision-maker preferences. The issue of the number of alternatives to be used for the approximation is discussed. The resulting model for the utility function can be used in order to avoid the formulation of new queries to the decision-maker in new instances of the same decision-making problem.

2 Utility Function Approximation

In this paper, our goal is to find a function $\hat{U}$ that models the decision-maker preferences (the utility function U). The role of such a function is to allow the replacement of the decision-maker in situations in which several decision problems will be presented, considering at each time a different subset A_i of available alternatives, from the set $\mathcal{A}$ of all possible alternatives.

The utility function U, which represents the decision-maker preferences and assigns a scalar value to each alternative, is initially unknown. The aim is to build a representation $\hat{U}$ of U that preserves the ordinal relationship between any two points – which is equivalent to the statement that the level sets of U and $\hat{U}$ must be the same. It should be noticed that the actual values of U and $\hat{U}$ become irrelevant.

The problem of finding $\hat{U}$ can be stated as a regression problem, that should be performed over sampled points coming from U. However, as only ordinal information can be obtained from U, a partial ranking procedure is employed here in order to allow such a regression.

The Kendall-tau distance (KTD) [9] is used here as a merit function to evaluate the approximated function $\hat{U}$. The KTD is a measure of proximity between the sorting for the alternatives provided by the approximation $\hat{U}$ and the ideal sorting provided by utility function U. In the case of the study conducted here, the KTD was suitable because an absolute reference (the U model) was assumed to be available in the tests that were performed.

The proposed method consists of three main steps:

Step 1: Choose a domain $\mathcal{A}$ for approximation;
Step 2: Build a partial ranking, assigning a scalar value to each alternative and finding a partial sorting for the alternatives;
Step 3: Construct an artificial neural network $\hat{U}$ which interpolates the results and approximates the decision-maker utility function U.

The next subsections present the details of those steps.

2.1 Step 1

The domain for $\hat{U}$ is inferred from the domain of the instance i of the decision-making problem, i.e., from the available alternatives A_i. The domain is defined as the box constructed considering the minimum and maximum values for the available alternatives A_i in each problem dimension.

In this domain, a fictitious decision-making problem is built, in which the alternatives are located as a grid. The queries to the DM are presented over this grid. The grid is constructed to find a representation for the utility function U in the desired domain. The number of alternatives in the grid is related to the quality of the approximation $\hat{U}$: a fine grid provides a better approximation, but, in this case, many queries will be asked to the DM.

2.2 Step 2

The partial ranking is a technique used to find a partial sorting for the alternatives, assigning a scalar value to each alternative. Considering a set $\mathcal{A}$ with n alternatives, this process is performed through the following steps:

- Choose randomly $p = \log n$ alternatives[1] from set $\mathcal{A}$; these alternatives are called the *pivots*. This value of p is chosen by inspiration in traditional sorting procedures. Moreover, this value insures enough data for the regression technique.
- Sort the pivots in ascending order of DM preferences. A rank is assigned to each pivot, corresponding to its position in this sorted list. This sorting is performed using ordinal information obtained from yes/no queries. The number of queries that the decision-maker has to answer is equal to the number of comparisons that a sorting algorithm would have to make. Therefore, using a algorithm like quicksort, the average number of queries to obtain a total ordering is $p \cdot \log p$.
- For each $n - p$ remaining alternatives, assign a rank that is the same of the pivot immediately better than the alternative, in the DM preference. Each alternative is compared with the middle pivot and, based on the result, compared with the middle pivot of the lower or higher subpartition. This process continues until a rank is assigned. If the current alternative is better than the rank p pivot, it receives rank $p+1$, and p is increased. The number of queries to obtain a rank for all remaining alternatives is $(n - p) \cdot p$.

This procedure creates a partition of the set $\mathcal{A}$ in at least p disjunct subsets. As the number of pivots is less than the number of alternatives, many alternatives will have the same ranking, providing a partial sorting. A total sorting could be obtained through a total ranking, but these results would be useless for the purpose of building a regression model using ANNs. For this reason, a partial sorting is strictly necessary.

All the comparisons between the alternatives and the pivots are obtained from ordinal information. With n alternatives, the number k of comparisons (queries to the DM) for performing this procedure is $O(n \cdot \log n)$ and approximately given by:

$$k \approx \log n \cdot \log(\log n) + (n - \log n) \cdot \log n. \tag{1}$$

The ranking-based classification offers a quantitative (cardinal) way to compare the alternatives, a kind of information which is not provided directly by the decision-maker. In any case, an alternative which is assigned a level $i+1$ is necessarily better than an alternative with a level i, although two alternatives with the same level i may be not equivalent under the utility function U.

2.3 Step 3

In this paper, the regression tool chosen is an artificial neural network (ANN). A ANN is an information processing paradigm which is inspired by the way information processing mechanisms of biological nervous systems, such as the brain, process the information. The key element of this paradigm is the structure of the information processing system, which is composed of a large number of

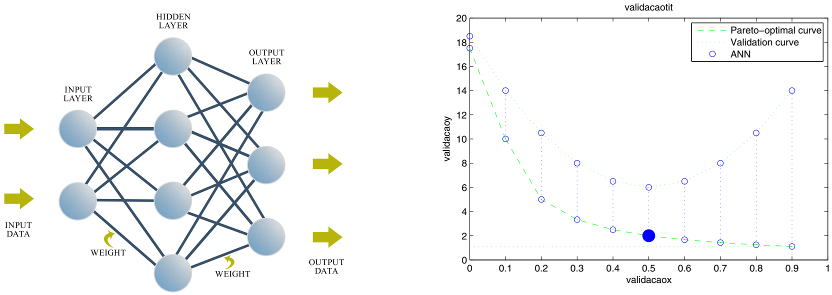
[1] We use $\log x$ as the same of $\log_2 x$.

highly interconnected processing elements (neurons) working together to solve specific problems.

The ANN learns by examples. The objective of this learning is the attainment of models with good generalization capacity, associated to the capacity to learn from a reduced set of examples and to provide coherent answers to unknown data.

The ANN architecture chosen for the application was a multilayer perceptron (MLP), with one hidden layer with 30 neurons. We use a multi-objective optimization approach to balance the training data error and the weight vector norm, to avoid underfitting and overfitting. The selection of the most appropriate solution within the Pareto-optimal set is performed by minimum validation error [3,16].

Figure 1 presents the architecture and an example of a Pareto-optimal set for an ANN training and the validation error for each solution found by the multi-objective algorithm. This curve has a minimum error, providing the detached solution for the ANN approximation.



Architecture Decision making

Fig. 1. Multilayer perceptron (MLP)

We use the alternatives within the grid as input and the ranking level of each alternative as output to train the ANN $\hat{U}$ which approximates U. It is not necessary to model U exactly, because the partial ranking keeps the partial sorting of the alternatives. When the ranking is used to build the approximation, we find a function with level sets which are similar to the ones of U and that possesses information enough to codify the decision-maker preferences.

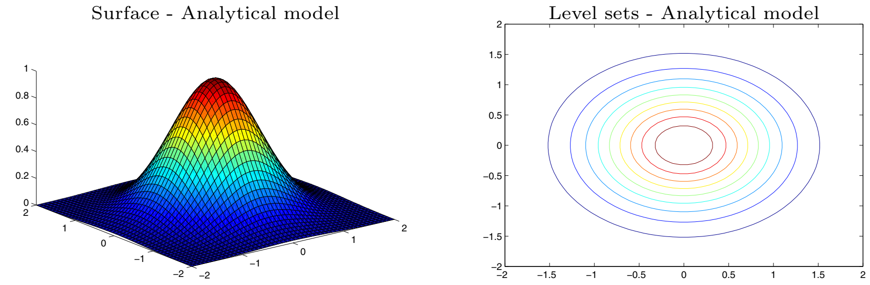
The algorithm 1 presents the pseudocode for the proposed method.

3 An Illustrative Example

Considering $p = (a, b)$ an alternative, we choose a bidimensional Gaussian (see Figure 2) to represent an analytical model for U. This simplified model, given by $e^{-(a^2+b^2)}$, will be used to simulate the decision-maker preferences. The domain

Algorithm 1. Proposed method - Pseudocode

```
1: Find the domain
2: Construct the grid of alternatives
3: Select the pivots
4: Sort the pivots in ascending order
5: Assign a rank to each pivot
6: Assign a rank to each $n - p$ remaining alternatives
7: Train the artificial neural network
```



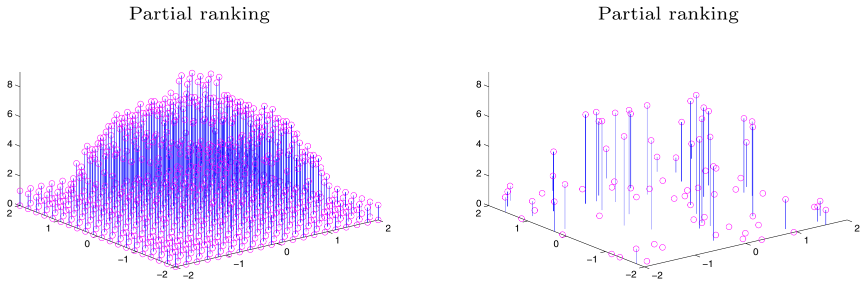
Surface

Level sets

Fig. 2. Simulated underlying model for the decision-maker preferences

found by Step 1 is the $[-2, 2] \times [-2, 2]$ set. In this domain, we construct a grid of 400 fictitious alternatives, which simulates a decision making problem for the approximation.

This model was chosen because a simple preference function is likely to be unimodal, the preference value for each alternative must be non-negative and should decay to zero for bad alternatives. So, the Gaussian has the structure of a simple preference function similar to some ones that are expected to appear in practice.



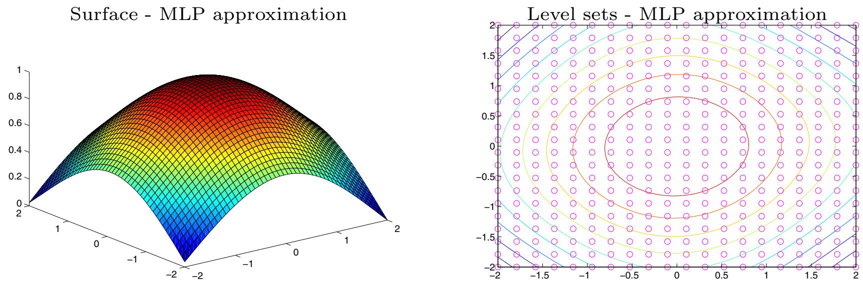
All the alternatives

10% of the alternatives

Fig. 3. Partial ranking

Figure 3 presents the partial ranking for the alternatives obtained by Step 2, as described in Section 2.2. Considering 30 runs, the average number of queries to find this partial ranking was 1285. For a better visualization, we present the same ranking with 10% of the alternatives.

Figure 4 presents the normalized surface and level sets of function $\hat{U}$, modeled by the ANN. Considering the grid of alternatives and creating 30 sets, each one with 50 random alternatives, the KTD average value in 30 runs was 0.02. These sets were created to find the KTD value in different sets, beyond the training set. The KTD value, 0.02, means that only approximately 2% of the queries made to the model have answers which are different from the ones of the analytical model. The $\hat{U}$ level sets approximate well the level sets of the utility function U, providing a way to solve the decision making problems without new queries to the decision-maker. Notice that the actual surface of the function $\hat{U}$ is rather different from the original utility function U. This difference is irrelevant under the viewpoint of generating correct ordinal comparisons among the points – the relevant information is in fact contained in the function level sets.



Surface

Level sets

Fig. 4. ANN approximation from grid of alternatives

4 The Number of Alternatives

An obvious problem when we use the grid of the alternatives is the exponential growth of the number of queries presented to the decision-maker, when the problem dimension grows. We performed the following experiment in order to find an indication about the number of alternatives that should be used in different problem dimensions, to discover an accurate approximation for the decision-maker preferences.

Considering an utility function U continuous and unimodal (as the Gaussian in the Figure 2), we create fictitious decision-making problems to find the behavior of the approximation procedure in relation to the number of alternatives and problem dimension. The domain of each fictitious decision-making problem is a hypercube with edge size 2 and the alternatives, distributed uniformly in this domain, will be considered for building the partial ranking.

Figure 5 presents the results obtained for problem dimension equal to 3, in which the axes are the number of alternatives in the fictitious decision problem and the KTD value. An approximation curve is constructed from the data regression to a logistic model given by $f(x) = \alpha_1.e^{\alpha_2+\alpha_3 x} + \alpha_4$. The suitable number of alternatives, n_d, is obtained considering a tolerance equal to 0.0001 in this regression. For different dimensions, the following numbers were obtained: $n_2 \approx 110$, $n_3 \approx 360$, $n_4 \approx 450$, and $n_5 \approx 720$.

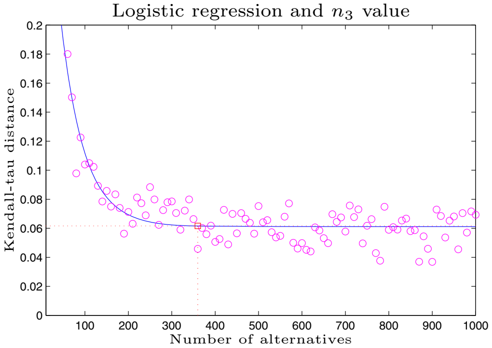


Fig. 5. Suitable number of alternatives

The number of queries to the DM varies according to the expression (1). Considering the maximum number of alternatives presented, the growth is almost-linear. Of course, the number of queries in the proposed approach is greater than the one expected in usual decision-making problems, under the paradigms of MAUT and OR. This is the price that should be paid for the ability to represent eventual dependencies (even non-linear ones) among the decision criteria for all of instances of a decision-making problem. It should be also noticed that the idea here is to get such needed information incrementally, possibly at the same time the DM takes decisions about specific instances of the decision problem.

5 Rank Reversal

The rank reversal problem consists in an inversion in the sorting of the alternatives provoked by the addition or subtraction of some alternatives. It is natural to hope that the addition or subtraction of some significant alternative modifies the decision-maker preferences and, consequently, the final results, but it is not intuitive that the insertion or removal of bad alternatives should provoke inversions in the sorting of the best alternatives. This problem was reported in AHP [10], in Promethee [11], and in Electre II and III [17].

Considering the proposed method, the training procedure of the ANN can invert the sorting of the alternatives, because the ANN is a regression method and, therefore, may represent data with some error. However, the inversions are

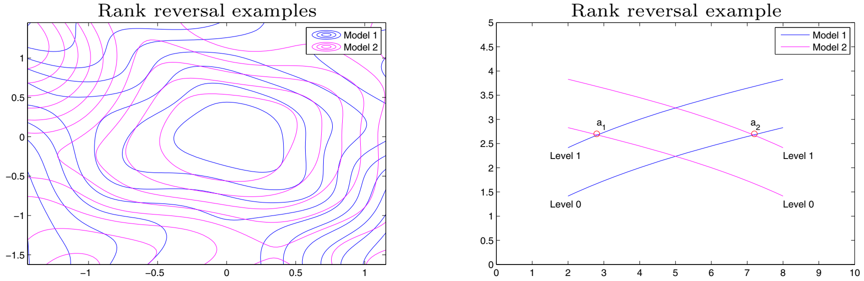


Fig. 6. Alternative a_1: level 1 in model 1 and level 0 in model 2. Alternative a_2: level 0 in model 1 and level 1 in model 2.

expected to be local[2], as long as the function U under approximation is expected to be smooth.

Figure 6 presents an example of local rank reversal, in which the level sets of two resulting models are compared. For each crossover among the level sets, the possibility of some rank reversal arises. In this example, the decision-maker model 1 prefers the alternative a_1 in relation to a_2 while the decision-maker model 2 prefers the alternative a_2 in relation to a_1.

6 Noisy Decision-Maker

Consider now that the decision-maker sometimes provides wrong answers to alternative comparison queries. A noisy decision-maker could perform an arbitrary choice sometimes, which expresses the case in which the decision-maker is uncertain about some answers. The proposed method is still applicable in cases like this.

In order to simulate this situation, some runs were performed with a fixed probability of the preference order being changed in any query. Figure 7 presents the typical cases for solutions obtained with such probability set to 5%, 10% and 20%, respectively.

As the ANN is a regression method, it can be seen that the noise affects the obtained results smoothly. This means that the ANN preference model suffers only a small perturbation for each new wrong DM answer, with the accumulated perturbation of several wrong answers causing a continuous variation of the model. This property makes the proposed procedure fundamentally different from other usual decision-making procedures, which would not be robust under such kind of uncertainty.

[2] In the sense that alternatives with similar preferences are expected to have rank reversal, but alternatives with very different preferences are not expected to suffer it.

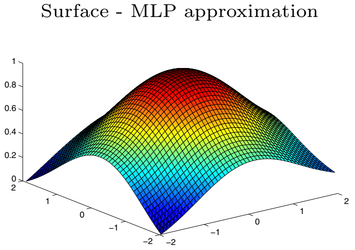
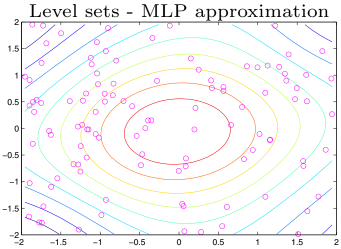
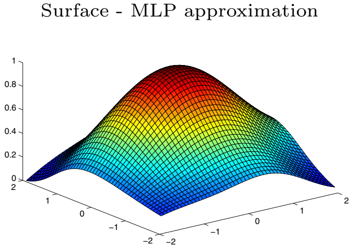
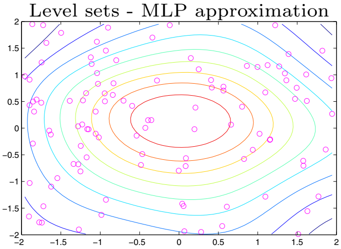
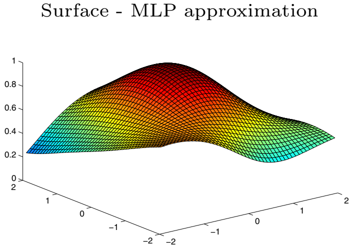
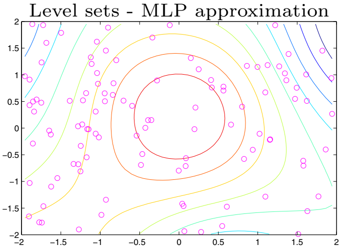
Noise	ANN approximation	ANN level sets
5%	 Surface - MLP approximation 	 Level sets - MLP approximation
10%	 Surface - MLP approximation 	 Level sets - MLP approximation
20%	 Surface - MLP approximation 	 Level sets - MLP approximation

Fig. 7. ANN approximation with noise

7 Conclusions

This paper proposed a methodology for the the construction of a function that approximates the decision-maker preferences using a partial ranking procedure and an artificial neural network. This function approximates the decision-maker preferences in a specified domain, in which the ANN is trained. This method is suitable when the usual assumptions of multiattribute utility theory about the decision-maker preferences are satisfied.

The problems in which the same decision-maker is consulted many times in repeated similar decision-making problems can be solved in the following way:

- Given a decision making problem, a partial ranking is built, leading to a partial sorting for the alternatives;
- With a set of alternatives and the partial ranking process, the resulting function is obtained by the ANN.

With this approximation, no more queries to the decision-maker are necessary in further instances of the same decision problem.

A feature of the proposed methodology is the ability of the method to represent arbitrary dependencies among the decision criteria, including non-linear dependencies, in the context of situations in which the DM should evaluate a solution as a whole, instead of weighting the criteria which should be used in order to evaluate a solution. The outcome of the proposed method has the purpose of being a representation of the DM preference structure in a region of the decision variable space, instead of being oriented to solve a specific decision problem.

Some favorable features of the proposed method are: (i) the inclusion/exclusion of information provides only local rank reversals instead of large scale ones that may occur in several current methodologies; and (ii) the proposed method is also robust to relatively large levels of wrong answers of the decision maker.

The authors are currently working in the application of the proposed method in some real-world problems.

References

1. Angilella, S., Greco, S., Lamantia, F., Matarazzo, B.: Assessing non-additive utility for multicriteria decision aid. European Journal of Operational Research 158, 734–744 (2004)
2. Angilella, S., Greco, S., Matarazzo, B.: Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral. European Journal of Operational Research 201, 277–288 (2010)
3. Braga, A.P., Takahashi, R.H.C., Costa, M.A., Teixeira, R.A.: Multi-objective algorithms for neural networks learning. In: Multi-Objective Machine Learning, pp. 151–171 (2006)
4. Brans, J.P.: L'ingénièrie de la décision; Elaboration d'instruments d'aide à la décision. La méthode PROMETHEE. In: Nadeau, R., Landry, M. (eds.) L'aide à la décision: Nature, Instruments et Perspectives d'Avenir, Québec, Canada, pp. 183–213. Presses de l'Université Laval (1982)
5. Chen, J., Lin, S.: An interactive neural network-based approach for solving multiple criteria decision-making problems. Decision Support Systems 36, 137–146 (2003)
6. Edwards, W., Hutton Barron, F.: SMART and SMARTER: Improved simple methods for multiattribute utility measurement. Organizational Behavior and Human Decision Processes 60(3), 306–325 (1994)
7. Fishburn, P.C.: Additive utilities with incomplete product set: Applications to priorities and assignments. Operations Research (1967)

8. Keeney, R.L., Raiffa, H.: Decisions with multiple objectives: Preferences and value tradeoffs. J. Wiley, New York (1976)
9. Kendall, M.G.: A new measure of rank correlation. Biometrika 30(1/2), 81–93 (1938)
10. Kwiesielewicz, M., Van Uden, E.: Inconsistent and contradictory judgements in pairwise comparison method in the AHP. Comput. Oper. Res. 31(5), 713–719 (2004)
11. Li, S.: Rank reversal properties of multicriteria decision making models. Master's thesis, University of Birmingham (July 2008)
12. Miller, D.W., Starr, M.K.: Executive decisions and operations research. Prentice-Hall, Englewood Cliffs (1960)
13. Roy, B.: Classement et choix en présence de points de vue multiples: La méthode ELECTRE. Revue Francaise d'Informatique et de Recherche Opérationnelle 8, 57–75 (1968)
14. Saaty, T.L.: A scaling method for priorities in hierarchical structures. Journal of Mathematical Psychology 15, 234–281 (1977)
15. Saaty, T.L.: Decision Making with Dependence and Feedback The Analytic Network Process. RWS Publications, Pittsburgh (1996)
16. Teixeira, R.A., Braga, A.P., Takahashi, R.H.C., Saldanha, R.R.: Improving generalization of mlps with multi-objective optimization. Neurocomputing 35(1-4), 189–194 (2000)
17. Wang, X.: Study of ranking irregularities when evaluating alternatives by using some ELECTRE methods and a proposed new MCDM method based on regret and rejoicing. Master's thesis, Louisiana State University (May 2007)

A MCDM Model for Urban Water Conservation Strategies Adapting Simos Procedure for Evaluating Alternatives Intra-criteria

Marcele Elisa Fontana, Danielle Costa Morais,
and Adiel Teixeira de Almeida

Department of Management Engineering, Federal University of Pernambuco,
PO Box 7462, Recife - PE, 50.722-970, Brazil

Abstract. This paper proposes a MCDM model based on the SMARTER method for the problem of urban water conservation strategies. The main contribution of this decision model is the use of Simos procedure adapted for evaluating alternatives intra-criteria, specifically on qualitative attributes. In fact, analyses of water resources problems are normally very complex, not only because this involves multiple alternative actions and objectives, but also because many attributes are subjective or need to be evaluated qualitatively. Nevertheless, there are many approaches to dealing with a semantic scale so as to make evaluation easier. However, it is often the case that the DM does not feel comfortable with the fixed nominal scale adopted or with the number of evaluations since the alternatives often need to be compared pairwise. On the other hand, a simple conversion from a nominal to an ordinal scale normally loses information. For these reasons, this paper proposes an adaptation of Simos procedure, associating 'playing cards' to evaluate alternatives on qualitative attributes, in order to facilitate the elicitation process. To show the applicability of the model proposed, a Brazilian case study is conducted.

Keywords: Intra-criteria evaluation, Revised Simos procedure, Water Conservation, SMARTER.

1 Introduction

Many conflicts among water users have emerged because the increasing demands for water, resulting in water shortages. These are also due to urban pollution affecting the quality of water and thus its use in certain activities being prevented [1]. In view of this situation, there is a need to look for strategic alternatives for using water efficiently in urban areas so as to avoid exacerbating the problem.

Ahmad and Prashar [2] evaluated different water conservation policies for their potential to reduce urban water demand. They found that indoor water use can be reduced by using low flow appliances such as toilets, faucets, showers, and washers. Outdoor water use can be reduced by using native plants and

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 564–578, 2011.

water smart landscaping (i.e., xeriscaping). Also water pricing has the potential to reduce water demand both for outdoor and indoor use.

Another increasing awareness around urban water conservation is the high index of water losses in distribution network [3]. According to Morais and Almeida [4], leakage represents a significant portion of the water losses index and is one of the crucial issues to be dealt in order to improve the efficiency and effectiveness of water supply services.

Kallis *et al.*[5] stated that urban water conservation might include longer-term policies such as consumer education, retrofitting and price reform, as well as short-term responses, such as voluntary appeals for restraints in water consumption, mandatory cutbacks of certain uses or tariffs penalizing excessive consumption. They also reported that the determinants of conservation behavior are the socio-economic characteristics of the water users, such as income or types of uses. Besides, physical factors such as temperature and rainfall have been used to explain the levels of water consumption [6].

In this context, decisions in the field of water resources are very complex and involve multiple actions and multiple objectives [7], often with intangible consequences. These characteristics make a multi-attribute analysis of alternatives an interesting way to deal with this kind of decision problem [8].

Nevertheless, many attributes used in analyzing alternatives for urban water conservation are qualitative. When evaluating by means of the multi-attribute method, it is common to make use of a nominal scale (bad - good - excellent), and later change from a nominal to a numerical scale (0 - bad, 1 - good, and 2 - excellent) to obtain values, and thus, important information of cardinality is lost in this process, especially if the nominal scale used is small.

Shepetukha and Olson [9] state that existing multi-attribute methods attempt to infer human preferences based on exact statements and evaluations - regardless of whether the humans involved have a clear understanding of the questions that they are asked. In fact, it is known that many decision makers have difficulty in assessing alternatives on qualitative attributes. Weber [10] argued that a decision-maker's preferences are rarely well enough structured to allow most decision analysis methods to be successfully applied.

In that perspective, Bana e Costa and Vansnick [11] proposed the MACBETH approach as a simple questioning procedure to 'drive' the interactive quantification of values through pairwise verbal judgments of the difference of attractiveness between valuable elements of a set of alternatives. However, sometimes the decision maker (DM) can not feel satisfied with the semantic scale adopted, and also, the MACBETH questioning procedure can be exhausting as it asks the DM for an absolute verbal judgment about the difference of attractiveness between each ordered pair of the set of alternatives.

'The limitations in human ability to evaluate and to compare multi-attribute options can lead to inconsistencies in human judgments or to the application of simplified rules that do not consider essential aspects of the options under consideration' [12]. To avoid this problem, this paper proposes to use the revised Simos' procedure [13] adapted for evaluating alternatives on qualitative attributes.

Simos [14] proposed a technique that allows any DM - not necessarily familiarized with multicriteria decision aiding - to think about and express the way in which he wishes to rank the different criteria of a family F in a given context. Therefore, Simos' aim was to use the procedure as a way to facilitate the elicitation of criteria weights based on a 'card playing' procedure in which different criteria are classified in different subsets by the DM, followed by ranking and weighting the subsets. The method is simple and practical. Figueira and Roy [13] proposed the revised Simos' procedure which is similar to the original one in that it allows the DM to associate 'card playing' with directly converting his/her priorities into weights, but, as compared to the original version, it has some advantages in terms of: processing information to obtain non-normalized weights; and of minimizing the rounding of errors when normalized weights are calculated. Our goal, however, is to use the revised Simos procedure to elicit from the DM his/her assessment of alternatives within qualitative attributes, i.e., to obtain the evaluation intra-criteria in terms of performance of alternatives with a cardinal notion.

Given the evaluation of all alternatives by all relevant attributes, in order to achieve the expected result, i.e. what strategic alternative for urban water conservation should be undertaken, the SMARTER (Simple Multi-attribute Rating Technique Exploiting Ranking) method developed by Edwards and Barron [15] is used. This method assumes that the DM's set of weight information consists of ranked swing weights, i.e., a ranking of the importance of the attribute ranges, and in this context uses 'surrogate weights' derived from this ranking. The particular surrogate weights are called rank order centroid (ROC) weights [16].

SMARTER is an additive multi-attribute method, i.e. the method finds the sum of the utilities of the alternative under all attributes. Therefore, just as an error in the value of the evaluation of an alternative, in an attribute, this can cause inefficiency in the final result. It is evident that additive methods are more sensitive in assessments. This view increases the need for using more robust procedures to evaluate alternatives on qualitative attributes, rather than a simple nominal scale.

The paper is structured as follows. The second section provides information about the model proposed as a structure for the methodology, the revised Simos procedure adapted to evaluate alternatives and the SMARTER method. Section 3 presents the case study, including: a survey of alternatives and attributes; an application of the revised Simos procedure adapted to evaluate alternatives; and the results and analysis by SMARTER. Finally, Section 4 presents final considerations.

2 Proposed Model

The model proposed consists of five main steps, as presented in Fig. 1. The first is characterizing the problem, when surveys of the potential alternatives and the relevant attributes are conducted. In the second stage, the attributes are divided into two classes: qualitative and quantitative. In the third stage, for quantitative attributes, the evaluation of the alternatives performance per attributes is given, the entries for which should be scores or physical value-related measures; for qualitative attributes, the revised Simos' procedure adapted to evaluate alternatives is applied. When the sum of the values of the evaluations of alternatives is not 100 points, a review is made of the values, in which the set of alternatives is divided, by calculation (defined later in this paper), into two lists: F^+ (rounded up), and F^- (rounded down).

The fourth stage of the model consists of implementing the SMARTER method, by evaluating alternatives by attributes. For this stage, all attributes are considered (qualitative and quantitative), from which the end result is obtained in the fifth stage, namely, the alternative with the best multi-attribute utility.

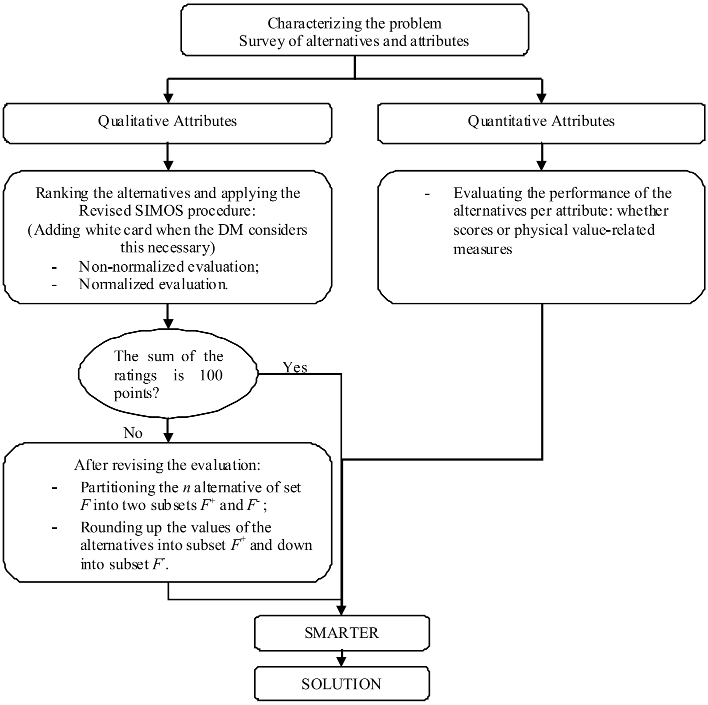

Characterizing the problem
Survey of alternatives and attributes
Qualitative Attributes
Quantitative Attributes
Ranking the alternatives and applying the Revised SIMOS procedure:
(Adding white card when the DM considers this necessary)
- Non-normalized evaluation;
- Normalized evaluation.
- Evaluating the performance of the alternatives per attribute: whether scores or physical value-related measures
The sum of the ratings is 100 points?
Yes
No
After revising the evaluation:
- Partitioning the n alternative of set F into two subsets F^+ and F^-;
- Rounding up the values of the alternatives into subset F^+ and down into subset F^-.
SMARTER
SOLUTION

Fig. 1. Structure of the model proposed

The model proposed will be applied in a case study for evaluating strategic alternatives for conserving and using water efficiently in urban areas. This study is presented later in this paper. Thereafter, the Revised Simos' procedure is summarized and adapted to evaluate alternatives on qualitative attributes, as is the SMARTER method.

2.1 Revised Simos Procedure - Adapted to Evaluate Alternatives

The Simos procedure uses a 'card playing' to facilitate the intuitive understanding of attributing numerical values to the weights of each criterion. The DM handles the cards, which represent each criterion, in order to rank them, and inserts white ones between subsets of criteria (criteria of the same importance). Each white card that the DM places between two successive subsets means one more scale unit of difference between their corresponding weights. The DM can insert as many white cards as he wants to represent the distance between two subsets of criteria. The revised Simos procedure to achieve the values of the criteria weights is available at [13].

In this study, instead of finding the criteria weights, we propose to apply the revised Simos procedure to obtain values for alternatives under qualitative attributes. Thus, an illustrative example is presented of the steps used to implement the methodology of the revised Simos procedure which was adapted to evaluate alternatives under qualitative attributes.

Consider a set A with 6 alternatives: $A = a, b, c, d, e, f$ which should be evaluated under the qualitative criterion C. Give the DM a set of cards with the name of each alternative written on each card. Therefore, there are n cards, n being the number of alternatives (6), and a set of white cards the number of which will depend on the DM's needs. Then, ask the DM to rank these cards (or alternatives) from the least to the most preferable (alternatives with the worst and the best evaluation). Some alternatives may be of the same evaluation to the DM (i.e., same value) and the DM can put them in the same subset (in this case, the subset is deemed to consist of *ex aequo* alternatives). Next, the DM introduces white cards between two successive cards of alternatives (or subsets of cards) to represent the distance between any two adjacent subsets (as shown schematically in Fig. 2). Therefore:

- No *white* card means that the distance is equal to a scale unit, u.
- One *white* card means a difference of *twice* u, and so on

Each white card placed by the DM between two successive subsets means one more scale unit difference between their corresponding values. Subsets of alternatives for further identification can be numbered from 1 to $\overline{n}$; where $\overline{n}$ is the number of subsets.

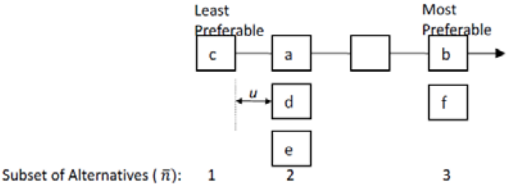


Fig. 2. Illustration of playing cards (ranked by DM)

Based on Fig. 2, the procedure to obtain the evaluation of the alternatives is undertaken. Each alternative is placed in a position in accordance with the rank of the subset to which it belongs (the least preferable alternative receives Position 1, the next one, Position 2, etc). The white cards are also positioned. However this is not included in the total sum of the positions. When there is more than one alternative in the same rank, the average of their positions needs to be used, resulting in a Non-normalized evaluation for these alternatives. The normalization evaluation is the ratio between the Non-normalized evaluation and the total sum of positions. The alternatives in the same subset are given the same evaluation. The final sum of the evaluation should result in 100 points. This procedure is presented in Table 1.

Table 1. Simos' procedure applied to evaluating alternatives

Subset of alternative	Number of cards	Position	Non-normalized evaluation	Normalized evaluation	Total
c	1	1	1	$\frac{1}{23} * 100 = 4.34 \rightarrow \mathbf{4}$	$1 * 4 = 4$
a, d, e	3	2,3,4	$\frac{2+3+4}{3} = 3$	$\frac{3}{23} * 100 = 13.05 \rightarrow \mathbf{13}$	$3 * 13 = 39$
White Card	1	(5)	-	-	-
b, f	2	6,7	$\frac{6+7}{2} = 6.5$	$\frac{6.5}{23} * 100 = 28.26 \rightarrow \mathbf{28}$	$2 * 28 = 56$
Sum	**7**	**23***	...	...	99

*The sum does not include the position of the white cards.

As seen in the example, sometimes the sum of the evaluation cannot total a hundred points: this may be lower or higher. Therefore, the revised Simos procedure needs to be applied, which determines the alternative that will be rounded up and those that will be rounded down, so that the sum total is 100 points.

The first step is to determine the ratio between the evaluations of the most and the least preferable alternative in the ranking. The ratio is equal to the total number of cards, T. But, if there is a subset with q most important alternatives and a subset with p least important alternatives in the ranking, the ratio is obtained from Eq. (1) [13].

$$z = \frac{[\sum_{i=0}^{q-1}(T-i)]p}{[\sum_{i=0}^{p-1}(1+i)]q} \tag{1}$$

As to the ratio z, the value u is calculated using Equation (2).

$$u = \frac{z-1}{e} \tag{2}$$

Where:

$e = \sum_{r=1}^{\overline{n}-1} e_r$ and $e_r = e'_r + 1$

e'_r is the number of white cards between the alternatives subsets r and $r + 1$, $(\forall r = 1, ..., \overline{n} - 1)$.

In this example, $z = 6.5$ and $u = 1.8333$, since $e = 3$. With this information, the non-normalized evaluation (k_r) can be calculated as shown in Table 2, where $k_r = 1 + u(e_0 + +e_{r-1})$ with $e_0 = 0$ and $K' = k_r$ (the number of alternatives in the subset $\overline{n}$).

Table 2. Non-normalized evaluation of alternatives

Subset of alternatives $\overline{n}$		Number of white cards between the subsets of alternatives r and r+1(e'_r)	$e_r = e'_r + 1$	Non-normalized evaluation (k_r)	Total(K')
1	c	0	1	1.00	1.00
2	a, d, e	1	2	2.83	8.49
3	b, f	—	—	6.50	13.00
sum	n=6	1	3	10.33	K=22.49

The second stage is to determine the normalized evaluation of each alternative i. Let k'_i be the evaluation of the alternative i in its non-normalized expression ($k'_i = k_r$). Considering $k^*_i = \frac{100}{K'} * k'_i$; and $k''_i = k^*_i$ with w figure after the decimal point ($w = 1$), it is necessary to create two lists L^+ and L^- defined as follows [13]:

- List L^+ is built by the pairs (i, d_i) ranked according to the increasing evaluation of the ratio $d_i.(d_i = [10^{-w} - (k_i{}^* - k''_i)]/k^*_i)$
- List L^- is built by the pairs (i, d^*_i) ranked according to the increasing evaluation of the ratio $d^*_i.(d^*_i = (k_i{}^* - k''_i)/k^*_i)$

With these lists, the n alternatives of set A are partitioned into two subsets F^+ and F^-, where $|F^+| = \nu$ and $|F^-| = n - \nu$. The alternatives of F^+ will be rounded upwards to the nearest whole number and the alternatives of F^- will likewise be rounded downwards. Since $\nu = (100 - k'') * 10^{+w}$ and the set $M = \{d_i > d^*_i\}$, $|M| = m$, the partition of F is performed as follows:

- If $m + \nu \leq n$, then construct list F^+ by the first ν alternatives of L^- not belonging to M; and F^- with the others.
- If $m + \nu > n$, then construct the list F^- by the $n - \nu$ last alternatives of L^+ not belonging to M; and F^+ with the others.

Based on the information above, the normalized evaluation in the example proposed is presented in Table 3. The numbers in brackets are used to identify all the alternatives belonging to the list F^+.

The normalized evaluation is the performance evaluation of each alternative on the attribute C, based on the DM's preferences. This evaluation is just in one qualitative attribute. This procedure must be applied to all qualitative attributes. After that, using all evaluations of qualitative and quantitative attributes, a multi-attribute method should be applied so as to find the solution to the problem.

Table 3. Normalized evaluation of alternatives

Subset of alternatives $\overline{n}$	Alternative	k'_i	k^*_i	k''_i	d_i		d^*_i	Normalized Evaluation
1	c	1.00	4.446421	4.4	0.012050	>	0.010440	4.4
2	a	2.83	12.583370	12.5	0.001322	<	0.006625	12.6 (3)
2	d	2.83	12.583370	12.5	0.001322	<	0.006625	12.6 (2)
2	e	2.83	12.583370	12.5	0.001322	<	0.006625	12.6 (1)
3	b	6.50	28.901734	28.9	0.003400	>	0.000060	28.9
3	f	6.50	28.901734	28.9	0.003400	>	0.000060	28.9
Sum			100	$k'' = 99.7$				100

2.2 SMARTER

The Simple Multi-Attribute Rating Technique Extended to Ranking (SMARTER) is based on SMARTS [15], but is simpler to use. Edwards and Barron [15] proposed SMARTS as a simplified version of MAUT (Multi-attribute Utility Theory), using linear approximations for single-dimension utility functions, an additive aggregation model and swing weights.

A key component while developing an additive multi-attribute value model is to obtaining the weights of attributes. SMARTER is an improvement on SMARTS because it makes it easier to elicit the weights of attributes. According to [15] SMARTER uses the same procedure as SMARTS except that it omits the second of two elicitation steps in swing weights, by substituting it with calculations based on rank. The rank, which represents the importance of the attribute ranges, is replaced by rank order centroid (ROC) weights [16]. The idea behind the centroid approach is to identify the weights with the minimum-maximum error from the extreme points implied by the rank of the attributes [17].

Edwards and Barron reported very little loss in accuracy in moving from SMARTS to SMARTER. They assert that a decision based on weights of the SMARTER method, on average, gains 98 to 99% of the utility obtainable by using full elicitation of weights [15].

SMARTER uses the same overall model as SMARTS, and cardinal input information where scores are standardized to a 0-1 scale (with 0 representing the worst expected performance on a given attribute, and 1 representing the best one).

Due to the fact that the weight assessment in SMARTER is simpler than in SMARTS, the former is used in the decision model proposed for evaluating strategic alternatives for urban water conservation. Especially when the decision maker has no knowledge about multi-attribute decision-aid approaches, it is very important to develop a decision making model based on a simple and easy way of using and understanding procedures, and this should have a theoretical foundation as well.

3 Case Study

Due to the problem of the world-wide shortage of water, several solutions are presented in order to combat wasting and misusing water in urban areas. Evaluating the alternatives for conserving water and using it efficiently in urban areas must be conducted by paying due regard to economic and financial, environmental, and social and technical matters. Moreover, the decision maker's preferences, personal values, points of views and knowledge about the problem should be taken into consideration.

To illustrate the application of the model proposed, a study was carried out in the region bounded by District-30 which is located in the district of San Martin in Recife, Pernambuco State, Brazil. The district has an area of 305.70 hectares, occupied by a population of 50,631 inhabitants. The urbanization is characterized as being from medium to low income with a focus on Poverty. Based on the characteristics of the region, the Decision-Maker, a manager of the State Water Resources company, made a survey of potential alternatives and relevant attributes, as presented below.

3.1 Survey of Alternatives

Given the current problem of water shortages in urban areas, it was possible to identify, by means of a literature review and interviews with experts in the field of water resources, some strategic alternatives that can be deployed as a way to alleviate or even solve the problem which this article discusses. The alternatives (strategies) are arranged in three categories:

- Actions aimed at reducing the consumption of water:
  - Strategy 1 (S1) - Reusing waste water;
  - Strategy 2 (S2) - Environmental education campaigns;
  - Strategy 3 (S3) - Setting up water saving devices;
  - Strategy 4 (S4) - Use of rainwater.
- Actions to ensure water quality:
  - Strategy 5 (S5) - Improving sewage projects in urban areas;
- Actions to increase operational efficiency in the water distribution network:
  - Strategy 6 (S6) - Controlling the Water pressure;
  - Strategy 7 (S7) - Monitoring fraud;
  - Strategy 8 (S8) - Controlling Leakages.

3.2 Survey of Attributes

Determining the list of relevant attributes was conducted along with the DM, and took into account economic, social, environmental and operational aspects under the current conditions of the problem of water shortages in urban centers, it being a matter of urgency to increase the conservation and efficient use of water and also the knowledge of agencies and organizations interested in the issue. Determining appropriate attributes is crucial for the efficiency and effectiveness of the method. The attributes determined are:

- Attribute 1 (A1) (initial investment): Corresponds to the monetary value invested in implementing actions (in R$).
- Attribute 2 (A2) (maintenance cost of the action): This refers to the monetary value invested in implementing action (in R$).
- Attribute 3 (A3) (response time of action implemented): This corresponds to the time needed to obtain the result of actions implemented (in months).
- Attribute 4 (A4) (impact of alternatives on the population involved): Corresponds to people's level of acceptance or rejection of action.
- Attribute 5 (A5) (efficiency action): Corresponds to verifying the results obtained by implementing the alternatives and that these are consistent with the objectives expected by the decision maker on a specific alternative.
- Attribute 6 (A6) (reducing consumption): Checks whether there has been a reduction in the volume of water consumed after implementing the action.

For the case under study, the attributes of initial investment (A1), maintenance cost of the action (A2) and response time of action implemented (A3) are evaluated according to estimates by national authorities and research in the area. The attributes are quantitative and the lower the value, the better the alternative. The other attributes (A4, A5 and A6) have a subjective character, i.e. these are qualitative attributes, to which the revised Simos procedure will be applied. It is important to notice that the attributes were chosen by the DM for the specific situation of the case under study. In other cases, other attributes can be analyzed, however should be observed their compensatory rationality.

3.3 Application of the Revised Simos Procedure to Evaluate Alternatives on Qualitative Attributes

The DM was asked to order the alternatives for each qualitative attribute. When finished this gave a ranking from the worst to the best alternative. The DM can insert white cards when the difference between the successive alternatives is more than one unit (the DM can insert as many cards as he wants to represent his evaluation of these alternatives.) The rank given by the DM for the alternatives on attributes A4, A5 and A6 are presented in Table 4.

Table 4. Ranking of the alternatives on the attributes A4, A5 and A6

Rank	A4	A5	A6
1	S6	S2	S2
2	S8	White card	White card
3	S5, S7	S1, S3, S4	S5
4	White card	White card	White card
5	White card	S5, S6, S7, S8	S6
6	S1, S4	—	S3, S7
7	White card	—	White card
8	S3	—	White card
9	White card	—	S1, S4, S8
10	S2	—	

Table 5. Normalized evaluations of attribute: A4

Rank	Alternative	k_i^*	k_i''	d_i		d_i^*	Normalized Evaluation
1	S6	2.17960	2.1	0.00936	$<$	0.03652	2.2 (6)
2	S8	4.83871	4.8	0.01267	$>$	0.00800	4.8
3	S5	7.49782	7.4	0.00029	$<$	0.01305	7.5 (3)
3	S7	7.49782	7.4	0.00029	$<$	0.01305	7.5 (4)
4	S1	15.49695	15.4	0.00020	$<$	0.00626	15.5 (1)
4	S4	15.49695	15.4	0.00020	$<$	0.00626	15.5 (2)
5	S3	20.83697	20.8	0.00303	$>$	0.00177	20.8
6	S2	26.15519	26.1	0.00171	$<$	0.00211	26.2 (5)
Sum		100	$K'' = 99.5$				100

Where: $z = 10$ and $u = 1.285714$

Table 6. Normalized evaluations of attribute: A5

Rank	Alternative	k_i^*	k_i''	d_i		d_i^*	Normalized Evaluation
1	S2	2.03046	2.0	0.03425	$>$	0.01500	2.0
2	S1	9.64467	9.6	0.00574	$>$	0.00463	9.6
2	S3	9.64467	9.6	0.00574	$>$	0.00463	9.6
2	S4	9.64467	9.6	0.00574	$>$	0.00463	9.6
3	S5	17.25888	17.2	0.00238	$<$	0.00341	17.3 (1)
3	S6	17.25888	17.2	0.00238	$<$	0.00341	17.3 (2)
3	S7	17.25888	17.2	0.00238	$<$	0.00341	17.3 (3)
3	S8	17.25888	17.2	0.00238	$<$	0.00341	17.3 (4)
Sum		100	99.4				100

Where: $z = 8.5$ and $u = 1.875$

Table 7. Normalized evaluations of attribute: A6

Rank	Subset of Alternative	Position	Non-normalized	Normalized Evaluation	Total
		evaluation	Evaluation	Evaluation	
1	S2	1	1	1.81	2.0
2	S5	3	3	5.45	5.0
3	S6	5	5	9.09	9.0
4	S3	6	6.5	11.81	12.0
4	S7	7	6.5	11.81	12.0
5	S1	10	11	20.00	20.0
5	S4	11	11	20.00	20.0
5	S8	12	11	20.00	20.0
Sum		55			100

After the DM has given the rank, the revised Simos procedure was applied. The sum of the normalized evaluation of alternatives under attributes A4 and A5 was 99 and 101, respectively, it being necessary to build the F lists by rounding them properly. The evaluation of the alternatives resulting from this procedure can be seen in Tables 5 and 6 for attributes A4 and A5. The sum of the normalized evaluation under A6 already resulted in 100 points, as shown in Table 7.

As seen, only the attributes A4 and A5 were necessary to revise the Simos procedure. Having found these values, it is possible to apply the SMARTER method, as shown below, so as to search for a final decision on the problem.

3.4 Results and Analysis by SMARTER Method

The first step toward using SMARTER is to define attributes and alternatives, and to eliminate dominated alternatives. Based on the characterization of the problem of water shortages in urban areas, previously presented, the alternatives-by-attributes matrix was built, as shown in Table 8. As can be noticed, there is no dominated alternative.

Table 8. Matrix of evaluation: alternatives-by-attributes

	Attributes					
Alternatives	A1	A2	A3	A4	A5	A6
S1	2,800,000	170,000	15	15.5	9.6	20
S2	36,000	5,000	6	26.2	2.0	2
S3	210,000	0	12	20.8	9.6	12
S4	1,500,000	100,000	24	15.5	9.6	20
S5	2,700,000	25,000	1	7.5	17.3	5
S6	55,000	11,500	2	2.2	17.3	9
S7	26,000	24,000	1	7.5	17.3	12
S8	25,000	24,000	1	4.8	17.3	20

Thereafter, to apply SMARTER, what is required is to rewrite the scores table - the output from evaluating the alternatives - to single-dimension cardinal utilities. To do so, the linearity was tested of single-dimension utilities for each quantitative attribute, in which scores are available. The other attributes were already elicited with cardinal utility. Also, conditional monotonicity was tested. Therefore, since the DM's preferences behave linearly on those attributes and the conditional monotonicity was verified, the additive model can be used.

Therefore, to proceed with SMARTER , it is necessary that all evaluations are on the same scale of measurement. For this reason, normalization is required; converting values of the alternatives between 0 and 1, where 0 is the worst alternative and 1 the best alternative on each attribute, whether its objective is to maximize or to minimize. Thereafter, the values of the weights were elicited from the DM: "If there is an alternative that had the worst score for all attributes examined, given the opportunity to exchange the evaluation in only one dimension, to change from the worst to the best value among the alternatives, which dimension would you improve?" The DM responded that it was attribute A6. This process was continued until all dimensions were ranked. Next, the following ranking of the attributes was obtained for the proposed case study: $A6 > A1 > A2 > A5 > A3 > A4$.

After ordering the attributes, SMARTER uses predetermined values called ROC weights (Rank Order Centroid weights) weights, thus simplifying the acquisition of multi-attribute utilities. In this case, $w_6 > w_1 > w_2 > w_5 > w_3 > w_4$, as follows: $w_6 = (1+1/2+1/3+1/4+1/5+1/6)/6 = 0.4083$; $w_1 = (0+1/2+1/3+1/4+1/5+1/6)/6 = 0.2417$; $w_2 = (0+0+1/3+1/4+1/5+1/6)/6 = 0.1583$;...; $w_4 = (0+0+0+0+0+1/6)/6 = 0.0278$.

Basically, SMARTER undertakes a sum of the relation between the evaluations of the alternative and weights of each attribute. So, for instance, for alternative 1 (Strategy 1 (S1)), the following expression is used for the global utility: $U_S1 = .4083u_{S1}(x_{S1A6}) + .2417u_{S1}(x_{S1A1}) + .1583u_{S1}(x_{S1A2}) + .1028u_{S1}(x_{S1A5}) + .0611u_{S1}(x_{S1A3}) + .0278u_{S1}(x_{S1A4})$.

Table 9 shows the results by the SMARTER method.

Table 9. Results of the SMARTER method

Alternative	U_{S_i}
S8	0.95
S7	0.77
S3	0.72
S6	0.71
S4	0.65
S1	0.50
S2	0.47
S5	0.38

The method presents alternative S8 as the best strategy (the greatest utility), for the case of the problem of conserving water in urban areas. This alternative has the best evaluation under attributes A1, A3, A5 and A6, where A1 and A6 are the greatest weights. In addition, it has third best rating on attribute A2, the third greatest weight. A4 is the attribute where the alternative has the worst performance, but, that attribute has the lowest weight for the DM. The second best alternative S7 has a similar performance if compared to S8. However, its evaluation on the qualitative attribute A6 is worse than for S8, exactly in the attribute with the greatest weight. As SMARTER is an additive method, in which the search for the best solution requires compensation (trade-offs) among the attributes to be evaluated, a simulation was conducted in which all attributes have equal importance for the DM. In this case study, alternative S8 persists as the winner. Other methods could be applied in this case, but SMARTER proved to be effective. Applying ROC weights was found to be representative of the DM's preferences regarding the ranking of attributes.

4 Concluding Remarks

Simos [14] proposed to associate 'playing cards' with eliciting criteria weights. Figueira and Roy [13] revised Simos' procedure to minimize the rounding off errors when the normalized criteria weights are calculated. In this paper, the latter approach was adapted to evaluate alternatives on qualitative attributes. The reason for this adaptation is due to the difficulties that some DMs have in evaluating alternatives under subjective aspects and also because DMs often do not feel satisfied with a simple nominal scale, a cardinality judgment being needed, nor do DMs not feel satisfied with the exhausting elicitation process used by existing methods.

This new approach to evaluating alternatives on qualitative attributes fits well with SMARTER, an additive method for multi-attribute utility measurement, which has the advantage of being simpler to use and elicits the weights of attributes. Thus, this paper put forward a decision-making model based on the SMARTER method, by applying the revised Simos procedure which was adapted so to evaluate alternatives on qualitative attributes.

The model proposed was applied in a case study to evaluate strategies for conserving water in urban areas. Three of the six attributes examined by the DM were qualitative and the revised Simos procedure which was adapted to evaluate alternatives was applied. It was verified that the adapted procedure is effective with respect to evaluating the alternatives, and gave greater flexibility to the DM during the process. In fact, the DM is not limited to any one kind of scale, which may increase or decrease the difference between the alternatives as deemed appropriate. In addition, the benefits of using the adapted procedure on qualitative assessments becomes more relevant if the attribute associated has a large weight which has been elicited from the DM.

The revised Simos procedure is simple and easy to use, requiring little computational effort, thus increasing its applicability. Furthermore, it is shown to be efficient when evaluating alternatives on qualitative attributes when applying an additive method, since in this kind of method, interval scale evaluation is required. As to future studies, the extension of the model proposed in cases of group decision making should be assessed by analyzing the characteristics of the process and its applicability, since in group decision making, the members usually have differing points of view.

Acknowledgements. This work has been partially supported by CNPq (Brazilian Research Funding Bureau).

References

1. Silva, V. B. S., Morais, D. C., Almeida, A. T.: A Multicriteria Group Decision Model to Support Watershed Committees in Brazil. Water Resources Management (2010) doi:10.1007/s11269-010-9648-2
2. Ahmad, S., Prashar, D.: Evaluating Municipal Water Conservation Policies Using a Dynamic Simulation Model. Water Resources Management 24, 3371–3395 (2010)
3. Morais, D.C., Almeida, A.T.: Water network rehabilitation: a group decision-making approach. Water S.A. 36, 1–7 (2010)
4. Morais, D.C., Almeida, A.T.: Group Decision-Making for Leakage Management Strategy of Water Network. Resources, Conservation and Recycling 52, 441–459 (2007)

5. Kallis, G., Ray, I., Fulton, J., McMahon, J.E.: Public Versus Private: Does It Matter for Water Conservation? Insights from California. Environmental Management 45, 177–191 (2010)
6. Randolph, B., Troy, P.: Attitudes to conservation and water consumption. Environmental science & policy 11, 441–455 (2008)
7. Morais, D.C., Almeida, A.T.: Water supply system decision making using multicriteria analysis. Water S.A. 32, 229–235 (2006)
8. Hajkowicz, S., Higgins, A.: A comparison of criteria analysis techniques for water resource management. European Journal of Operational Research 184, 255–265 (2008)
9. Shepetukha, Y., Olson, D.L.: Comparative Analysis of Multi-attribute Techniques Based on Cardinal and Ordinal Inputs. Mathematical and Computer Modeling 34, 229–241 (2001)
10. Weber, M.: Decision making with incomplete information. European Journal of Operational Research 28, 1–12 (1987)
11. Bana e Costa, C.A., Vansnick, J.C.: MACBETH - An Interactive Path Towards the Construction of Cardinal Value Functions. International Transactions in Operational Research 1, 489–500 (1994)
12. Figueira, J.R., Greco, S., Ehrgott, M. (eds.): Multiple Criteria Decision Analysis: State of the Art Surveys. Springer-Media, New York (2005)
13. Figueira, J.R., Roy, B.: Determining the weights of criteria in the ELECTRE type methods with a revised Simos procedure. European Journal of Operational Research 139, 317–326 (2002)
14. Simos, J.: L'évaluation environnementale: Un processus cognitif négocié. Thése de doctorat, DGF-EPFL, Lausanne (1990)
15. Edwards, W., Hutton, B.F.: SMARTS and SMARTER: improved simple methods for multi-attribute utility measurement. Organizational Behavior and Human Decision Processes 60, 306–325 (1994)
16. Barron, F.H., Barrett, B.E.: The efficacy of SMARTER - Simple Multi-Attribute Rating Technique Extended to Ranking. Acta Psychologica 93, 23–36 (1996)
17. Olson, D.L.: Comparison of three multicriteria methods to predict known outcomes. European Journal of Operational Research 130, 576–587 (2001)

A Multicriteria Decision Model for a Combined Burn-In and Replacement Policy

Cristiano Alexandre Virgínio Cavalcante

School of Engineering, Centre for Technology and Geosciences,
Department of Production Engineering, Federal University of Pernambuco,
Recife PE, Caixa Postal, 5125 CEP: 52.0 70-970, Brazil
cristianogesm@gmail.com

Abstract. This paper considers a multicriteria model for a combined burn-in and replacement process for a simple system comprising a single component from a heterogeneous population, consisting of two sub-populations that possess different failure characteristics. There are several papers that have dealt with mixed distributions. Nevertheless, suitable procedures for dealing with the distinct failure behaviours from these two sub-populations are limited. Furthermore, some combined policies of burn-in and replacement have not achieved consensus on their efficiency. Therefore, we consider a multicriteria model for supporting the decision-maker in a combined burn-in-replacement policy. This model enables the decision-maker to set up a combined burn-in-replacement policy by taking advantage of the broader view provided by a simultaneous evaluation of cost and post-burn-in reliability while also providing the possibility of inserting the decision-maker's preferences into the model. A case study is used to illustrate some advantages in comparison with results from the classical model (minimum cost).

Keywords: combined policies of burn-in and replacement, multicriteria model, mixture.

1 Introduction

According to Jiang and Jardine [1], a typical mixture consists of a heterogeneous population, in which the failures can be divided into early failures and wear-out failures. For this situation, where the time to failure of an item follows a mixture distribution, the failure behaviour is not trivial because a mixture of two distributions might produce different types of failure rate functions that describe different failure behaviours [2], [3]. In this context, it is not obvious how to determine a suitable procedure for reducing the early-phase failures while simultaneously dealing with the operational failures caused by wear-out.

While dealing with early failure and operational failure due to wear-out separately is appealing in practice, it is a large problem for equipment suppliers in charge of maintenance actions. If operational failures occur, suppliers have to compensate for the damages by providing spare parts or giving discounts on the

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 579–593, 2011.

overall cost of the contract to ensure exclusivity of supply for the system. Thus, potential early failures that might arise during operation must be eliminated. Moreover, the replacement age should be adjusted to avoid operational failures due to wear-out. Therefore, a combined process is required so as to approach the two different behaviours that emerge from the heterogeneous population. Management of early or wear-out failures requires that the supplier make decisions to achieve the best benefit from the partnership. The choices of suppliers or contract planning are different problems that are not discussed here. For more details about these problems, see [4].

A common process used to eliminate early failures is the burn-in test. According to Cha et al. [5], the burn-in technique is a method used to eliminate initial failures in field use. This method consists of running an item or system prior to its use in the field. This process is important in identifying weak items or avoiding early failures in the field. However, the main problem encountered in the burn-in process is determining how long the process should take because, besides the cost involved, this process may damage the component or may be inefficient. Therefore, many models that provide optimal burn-in times have been proposed by various researchers. The optimal burn-in time is generally obtained from an optimisation process that involves one performance criterion. According to Cha et al. [5], besides cost, various criteria have been used to define the optimal burn-in time, such as the mean residual life [6,7,8], the reliability of performing a given mission [6]; [8], or availability [5].

Despite the fact that there are different aspects of measuring the effectiveness of a burn-in process, these criteria are used individually, even if a simultaneous observation of more than one criterion (a multicriteria approach) might enrich the problem of determining the best time for burn-in. On the other hand, for maintenance in the general sense, some interesting studies using a multicriteria approach have recently appeared with regard to very different problems.

Besides the need to simultaneously deal with more than one criterion, other considerations must be taken into account. In some practical situations, the management of early failures is not enough to provide effective results. According to Jiang and Jardine [1], most papers about burn-in have emphasised applicable conditions in which, the item initially has a high rate of decreasing failure, and the weak sub-population represents a very small proportion of the entire heterogeneous population. In contrast, most practical problems do not correspond to these conditions. Instead, the weak sub-population has a longer lifetime than those that are usually considered, and its proportion is not very small. Therefore, rather than focusing only on the process of burn-in, an integrated process is more appropriate. This integrated process consists of a combined policy of preventive replacement and the burn-in procedure [9]. Jiang and Jardine [1] proposed combined preventive maintenance policies associated with the mixture model. Their model is based on two different decision variables: the duration of the burn-in process b and the age of replacement t (b+y). The objective of these models is to find optimal solutions for a combined burn-in-replacement policy to take into consideration the change in ageing behaviour of items from a heterogeneous population represented by a mixed distribution. The cost advantage of the combined policy over separate policies of burn-in and replacement is quite small [9]. However, the performances of other criteria indicate that preventive replacement is more effective in the combined policy. Moreover, a multicriteria approach could increase the advantages of this policy [1].

As a result, this article proposes a multicriteria model for a burn-in-replacement policy to support the decision-maker in choosing the most appropriate times for b and y, simultaneously taking into account not only two criteria but also the decision-maker's preference regarding these criteria. This article has six sections. In the first and second sections, we highlight the concept of a combined burn-in and replacement policy and mixed distributions, respectively. In the third section, we present some concepts of Multiple-Attribute Utility Theory (MAUT). Then, we specify the multicriteria model for a combined burn-in and replacement process. The model is illustrated using realistic data. Finally, we draw some conclusions from the study.

2 Mixed Distributions

In the literature, mixed distributions (see Fig. 1) have been studied by a substantial number of researchers [10,11,12]. This popularity is due to the fact that these mixtures might be used to model real situations in which some observation characteristics arise from a non-homogeneous population.

The time to failure is assumed to arise from a mixture distribution:$F(t) = pF_1(t) + (1-p)F_2(t)$, where p is the mixing parameter. Thus, the heterogeneous nature of the components implies a mixed distribution for time to failure.

Despite the flexibility in modelling some situations, the interpretation of failure-rate behaviour for even the simplest of mixtures is not trivial. As was

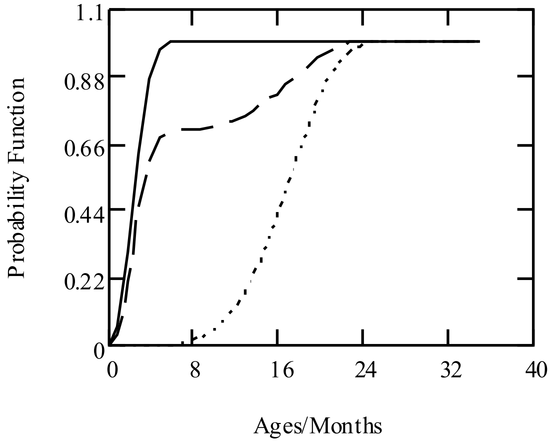


Fig. 1. Mixed distribution (__ __). Underlying Weibull distributions, $Weibull_1(\eta_1 = 18, \beta 1 = 5)$ (...) and $Weibull2(\eta_2 = 3, \beta_2 = 2.5)$ (___), mixing parameter $p = 0.7$.

stated by Barlow and Proachan [13], mixtures that result from Increased Failure Rates (IFR) are not necessarily IFRs. This aspect has produced some misunderstandings [11]; [3], and it represents a difficult barrier when it comes to proposing suitable maintenance policies for simultaneously reducing early-phase failures and operational field costs [1]. Finkelstein & Esaulova [14] explained why the mixtures of IFR distributions can decrease, at least for some intervals of time. These authors highlighted the aspect that the operation of mixing can change the corresponding pattern of ageing, which must be taken into account for practical purposes, especially in maintenance planning.

Therefore, in this paper, we focus on constructing a decision aid model for the maintenance planning of mixed items to take into consideration changes in the ageing behaviour of these items in a heterogeneous population. In practice, the main contribution of this paper is to help the decision-maker, who both supplies and provides maintenance service of a component in an exclusive contract, to determine the length of the burn-in process and the replacement age, to provide a feasible cost and good reliability in the field.

3 MAUT

The very nature of multiple criteria problems means that there is much information of a complex and conflicting nature, often reflecting differing viewpoints [15]. This naturally leads to associating a specific criterion with each pertinent point of view. Each of these criteria is used to evaluate any potential action on an appropriate qualitative or quantitative scale. Thus, the main problem encountered when dealing with diverse criteria lies in the aggregation of these different measures, taking into account the decision-maker's preferences and uncertainties present in the criteria. To manage these main difficulties, MAUT (Multiple-Attribute Utility Theory) is considered one of the most appropriate approaches, especially for cases in which the criteria are related to aspects of uncertainty.

MAUT provides axiomatic foundations for choices involving multiple criteria; its main result consists of establishing a functional form of the utility function based on confirmation of some of the assumptions with regard to the decision-maker's preferences [16]. Even that other kind of multicriteria decision aiding methods could be considered (see, [16]), here the MAUT seems to be the most suitable, not only due to the rationality of the decision maker, but also due the limitation of outranking approaches whose the set of alternatives has to be discrete.

Thus, to simplify the assessment of multiple-attribute utility functions and to warrant a more consistent result, a process with five stages should be followed [17]: (1) Introduction of the terminology and ideas; (2) Identification of relevant independence assumptions; (3) Assessment of conditional utility functions; (4) Assessment of the scaling constants; and (5) Verification of consistency and reiteration.

The first step is the most direct. It consists of making sure that the decision-maker has a good understanding of each element used in implementing MAUT.

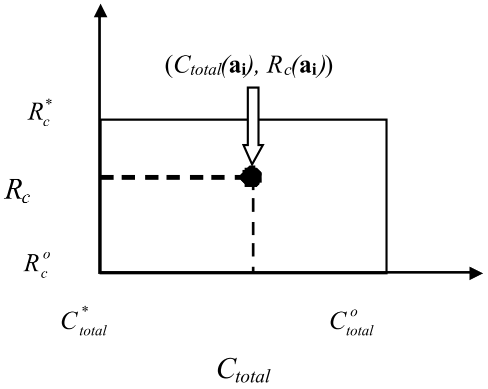


Fig. 2. Consequence Space

All subsequent steps rely mainly on the comprehension of the consequence space. For our particular case, the alternatives are represented by vectors of $\mathbf{a_i} = (b_i, y_i)$. For each alternative, there are consequences in terms of cost $C_{total}(b, y)$ and post-burn-in reliability $R_c(b, y)$. Thus, for each alternative $\mathbf{a_i}$, there is a corresponding associated point in the consequence space $(C_{total}(\mathbf{a_i}); R_c(\mathbf{a_i}))$; the range of each criterion is also established in the consequence space $[C^o_{total}; C^*_{total}]$ and $[R^o_c; R^*_c]$ (see Fig. 2).

The second step is very useful and consists of identifying a fundamental concept of multi-attribute utility theory, utility independence. When additive independence is observed, the representation of the multi-attribute utility function is reduced to a weighted sum of the single-attribute utility functions (see (1)). According to Fishburn [18], most of the applied work in multi-attribute utility theory deals with cases in which the multi-attribute utility function is broken down into its additive form.

$$U(C_{total}(\mathbf{a_i}), R_c(\mathbf{a_i})) = \sum_{j=1}^{N} k_j u_j(\mathbf{a_i}) \tag{1}$$

Where:

$u_j()$ is the single-attribute utility for each criterion C_{total} and R_c, respectively
k_j is the constant scale of a criterion
N is the number of criteria

The third step consists of breaking down the process of assessment. The utility function of each criterion is assessed separately. There are different processes for eliciting the utility function. Some are very complicated and comprise overly long questionnaires. Raiffa [19] proposed a very simple method that consists of determining some certainty equivalents when compared with some different lotteries to assess the utility function.

The fourth step involves assessing the scaling constants; this process depends on the final shape of the multi-attribute utility function. For the case of an additive form, there is a scaling constant for each criterion; in our particular case of two criteria, only an assessment of one scale constant is needed, determining the second constant by the equation $k_1 + k_2 = 1$ according to Keeney and Raiffa [17].

Finally, the fifth step consists of checking for consistency and reiterating. This step is very important in ensuring that the multi-attribute utility function is in fact connected with the decision-maker's framework. This process can be conducted in different ways [20].

4 Multicriteria Decision Model for a Combined Burn-In and Replacement Process

According to Cha [21], in a study of the burn-in problem, the criterion of cost is often adopted to find the optimal burn-in time. However, there are other circumstances in which certain system reliability characteristics such as system availability, probability of accomplishing a mission, mean life in field operation, etc. are more important than economic considerations; thus, in these cases, a multicriteria vision is quite useful.

In addition, instead of considering only the burn-in or replacement procedure, in an isolated way, we consider a multiple-attribute model of a combined burn-in-replacement policy.

As was stated by Drapella & Kosznik [9], for a non-homogeneous population of items, the dominant process of failure is related to wear-out. For this reason, a procedure of preventive maintenance is very suitable for this case. In spite of this, some items have a short life due to intrinsic defects. Therefore, according to these authors, when only an age replacement policy is undertaken and these weak items are used as replacement parts, they may spoil the results of planned prevention. Thus, there is a need to bring the burn-in and preventive replacement periods, respectively, b and t, into equilibrium. To do so, the multicriteria approach can provide a substantial enrichment of combined burn-in-replacement policies, especially because the cost advantages of this combined process are quite small. However, significant improvements in other aspects have been observed [9]; [1].

Based on a burn-in-replacement policy for a mixed distribution under the multicriteria decision aid paradigm and inspired by the model proposed by Jiang and Jardine [1], the multicriteria model presented here isolates the $R_c()$ aspect and transforms it into a decision criterion in addition to the $C_{total}()$ factor already used in classical models.

In fact, these two criteria are consistent with most problems involving burn-in and replacement. In general, there are two natural reasons for performing a burn-in [22,23,24]: burn-in offers a cost-effective opportunity to remove and replace defective items, while also being an important screening method used in predicting, achieving and enhancing field reliability. With regard to the age replacement policy, the matching of these criteria has been appropriately used in some papers [25,26].

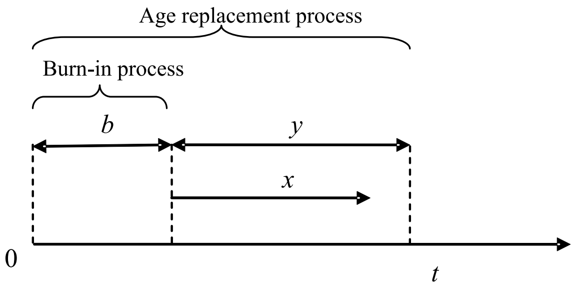


Fig. 3. Burn-in/replacement policy

As stated earlier, the population consists of two different kinds of items: weak items and strong items. They respectively present an earlier failure and a later failure. Therefore, a combined burn-in-replacement process is calculated with the respective times, b and y, in order not only to increase $R_c(b, y)$ but also to reduce the expected total cost per operational time $C_{total}(b, y)$.

Although the combined policy can deal with items with different failure behaviours, this process initially consists of a separation of the distinct processes: burn-in and replacement. In practice, basically all components must be submitted to a burn-in process, the duration b of which we want to identify. After a burn-in process, each item begins to operate and will be replaced at failure or if it achieves an age $t = b + y$ (see Fig. 3).

During the burn-in phase, the item will be repaired if it fails before the time b. Each repair consists of a renewal process, and the condition of "as good as new" is provided with cost C_r. There is also a cost-rate of operating the burn-in process, C_0. As is obvious, the associated cost and the time that an item should remain in burn-in depend on how often (m) the component fails before time b, see (2).

$$P(m) = R(b)F^m(b) \tag{2}$$

Thus, the expected value of $m(E(m))$ is directly obtained:

$$E(m) = sum_{n=0}^{\infty} mP(m) = F(b)/R(b) \tag{3}$$

The expected time of the burn-in process $E(t_b)$ is given by:

$$E(t_b) = \frac{1}{R(b)} \int_0^b R(t)dt \tag{4}$$

From (3) and (4), we can derive the expected burn-in cost:

$$C_b(b) = E(m)C_r + C_0 E(t_b) \tag{5}$$

For a burned-in item, an age replacement policy is undertaken as soon as the item fails or when it achieves the age $t = b + y$, as seen in Fig. 3. Therefore, the main problem is to establish both times b and y, which characterise the maximum operational time of the item at which point it should be preventively replaced. The failure distribution of x is also called the post-burn-in distribution.

$$G_b(b,x) = \frac{F(x+b) - F(b)}{R(b)} = 1 - R_c(b,x) \tag{6}$$

$1 - G_b(b,y)$ or $R_c(b,y)$ for a fixed b and $x = y$ represents the effectiveness of preventive maintenance [1] or the conditional reliability of a burned-in item. This is the probability that a burned-in item will be replaced at time t without failure, in which case this will be an attribute for the decision-maker.

Another attribute consists of the integrated cost-rate of the burn-in and replacement process ($C_{total}(b,y)$), which comprises the cost of the burn-in and the replacement process per unit of operational time.

$$C_{total}(b,y) = \frac{C_r F(b) + C_o \int_0^b R(t)dt + C_p R(b) + (C_f - C_p)[F(y+b) - F(b)]}{\int_0^y R(x+b)dx} \tag{7}$$

Unlike other studies [1,9], the multicriteria model establishes the times b and y by using MAUT. Thus, the main objective of this model is to determine the vector (b,y) that maximises the multi-attribute utility function of the decision-maker concerning the criteria $R_c(b,y)$ and $C_{total}(b,y)$; see ([1]).

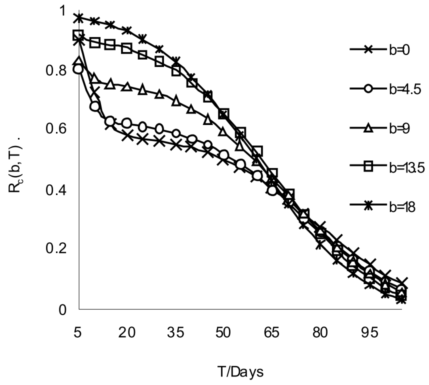


Fig. 4. Behavior of the Criterion $R_c(b,y)$ for different values of b

To understand the advantages that might be provided by a multicriteria model of a combined burn-in-replacement policy, we can observe the behaviour of the criteria $R_c(b,y)$ (Fig. 4) and $C_{total}(b,y)$ (Fig. 5). For the case in which $b = 0$, which is equivalent to a simple age replacement policy without burn-in, we can see that both the cost and reliability for most illustrative alternatives $\mathbf{a_i}(b = 0, y)$ are not so good in comparison with those obtained for other values of b. Even more interesting is the fact that there is no value of $\mathbf{a_i}$ that provides the best values of $R_c(\mathbf{a_i})$ and $C_{total}(\mathbf{a_i})$. Thus, the decision regarding the best policy does not consist of an optimisation process but of a multicriteria decision problem

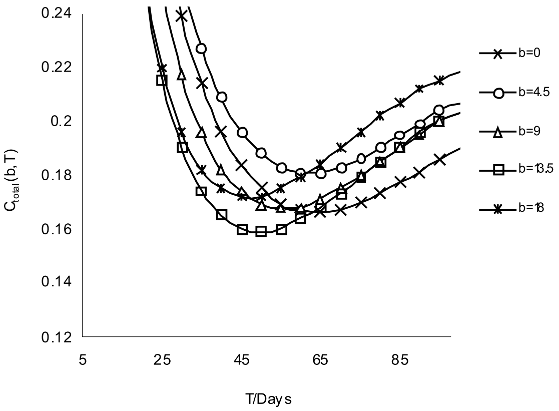


Fig. 5. Behavior of the Criterion $C_{total}(b, y)$ for different values of b

where not only the conflict of criteria must be overcome but where the decision-maker's preference has to be taken into account to define the alternatives that best match the decision-maker's desires.

5 Case Study

This application of a decision model is based on the context of a packing supplier for the juice and milk industries. The item subjected to burn-in is a component of the system provided by the packing supplier to the juice and milk industries.

Despite the fact that the system is not a simple one, we handle this system as a single component because the failures are almost all related to one part of this system; additionally, the maintenance contract of the supplier is related only to this component. The data correspond to realistic data. For this application, we consider that the failures of this component may arise from a heterogeneous population. Thus, we will assume a mixture with two component sub-populations: weak components and strong ones. In Table 1, we can see a summary of information about these components. In addition, reasonable values for the percentage of weak items may be around $p = 0.4278$.

Costs for this component were obtained subjectively from an expert. To more simply portray the impact of the costs on the combined policy, all costs are recorded relative to the cost of preventive replacement (C_p). Thus, the cost of preventive replacement is taken to be the unit cost (Table 2).

Table 1. Evaluation of the nature of the tasks

Weak		Strong	
Item	Weibull	Item	Weibull
	$\beta_1 = 1.98$		$\beta_2 = 1.98$
	$\eta_1 = 9.78$		$\eta_2 = 9.78$
	$p = 0.4278$		$1 - p = 0.5722$

Table 2. Evaluation of the nature of the tasks

Cost s	value
C_r	0.2
C_0	0.135
C_p	1
C_f	10

A combined burn-in-replacement policy was considered suitable for these items. However, the main problem is that there is no best alternative $\mathbf{a_i}$, that is, an optimal alternative for both criteria C_{total} and R_c. Thus, as stated earlier, from a multicriteria perspective, we want to find an alternative that best matches the decision-maker's preferences.

MAUT was the approach chosen to deal with uncertainty and conflicting criteria, as well as to take into account the decision-maker's preferences. In the following, we will describe the application of all of the steps of the MAUT method described previously.

Following the sequence for applying the MAUT (see Section 3), the range of each criterion was established. Thus, based on the behaviour of each criterion, we defined the consequence space as below (see Fig. 6). The set of alternatives A comprises all of the points (b, y) of the R^2 space. However, for the sake of applicability, we will consider the range of alternatives that are compatible with the mixed distribution function, which might be reasonable in a combined burn-in-replacement policy in practice. This includes the set of points relative to this space of consequences.

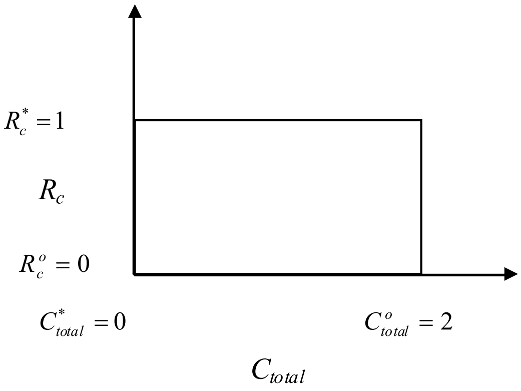


Fig. 6. Space of consequences

After verifying the independence assumptions regarding the criteria with the decision-maker, we confirm the additive independence. Thus, the final form of the function is very simple, as stated in Section 3.

Table 3. Mono-attribute utility functions

Criterion	Utility Functions	Parameters
C_{Total}	$u_1 = B_1 \exp(-G_1 * C_{total}(\mathbf{a_i}))$	$B_1 = 2.24$ $G_1 = 7.98$
R_c	$u_2 = B_2 \exp(-G_2/R_c(\mathbf{a_i}))$	$B_2 = 2.71$ G_2=1

To translate the values of the criteria to the same scale of preference, we must assess the respective utility functions for each criterion. There are different techniques for eliciting this kind of function from the decision-maker. Here, we used a simple and very useful approach [19], which results, with a good fit, in an exponential and logistic utility function for criteria C_{total} (Fig. 7) and R_c (Fig. 8), respectively (for details of these functions, see Table 3).

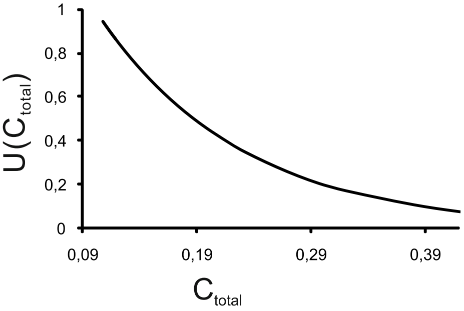


Fig. 7. Utility function for the criterion $C_{total}(U(C_{total}))$

Because the additive independence was confirmed and each single-attribute utility u_j was obtained in the final steps, to fully assess the multi-attribute utility function, we now have only to obtain the scale constants. Following a process explained by Keeney [20] for the assessment of scale constants, we obtained the following results: $k_1 = 0.55$ and $k_2 = 0.45$.

Finally, the multi-attribute utility function was determined (see (1)), allowing us to evaluate the set of alternatives A to rank or choose the best among them.

For this set of alternatives, we have the following values for the multi-attribute utility function $U(b, y)$ (see Fig. 9).

To complete the full MAUT application process, we have to confirm that the results provided by the multi-attribute utility function are consistent with the decision-maker's preferences. The consistency step is very important, as it might indicate the need to review one or more steps of the entire process.

The alternative with the highest value of $U(C_{total}(), R_c()) = 0.882$ was the vector $(b = 16.17, y = 55.91)$. This alternative corresponds to a burn-in time of 16.197 weeks and a replacement age of $(16.17 + 55.91)72.08$ weeks. If we

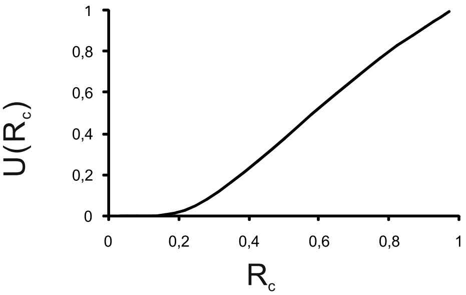


Fig. 8. Utility function for the criterion $R_c(U(R_c))$

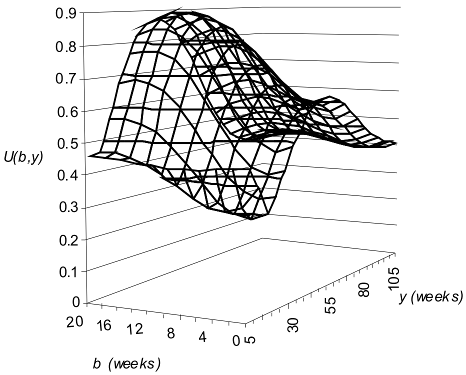


Fig. 9. Multi-attribute Utility function $U(C_{total}, R_c)$

evaluate the values provided by this alternative according to both criteria, we have: $C_{total}(b = 16.17, y = 55.91) = 0.109$ and $R_c(b = 16.17, y = 55.91) = 0.828$.

We can see that, in comparison with the best alternative obtained when only observing the classical criterion C_{total}, which consists of an optimisation process or a search approach seeking the vector that minimises this criterion, a significant difference is observed, especially for the R_c criterion. The optimum point is $(b = 14.314, y = 72.962)$. The performance of this point for both criteria corresponds to $C_{total}(b = 14.314; y = 72.962) = 0.102$ and $R_c(b = 14.314, y = 72.962) = 0.695$. As was expected, this point, in terms of costs, is slightly better than the alternative obtained by the multicriteria model. On the other hand, in terms of Rc, there is a clear improvement in the efficiency of preventive maintenance when the decision is made using a multiple-criterion approach.

We can observe that a combination of activities in maintenance procedures has produced much better results than a pure policy (see [27,28]) for the context of heterogeneous items. This seems to be a natural requirement for handling the complex failure behaviour of these mixtures. On the other hand, we can see, through the results provided by the multicriteria model, that some novel improvements can be introduced. Using the multicriteria model, the set of a specific policy can be driven to give results that best match the decision-maker's preferences.

It is worth to notice that the multicriteria decision aiding approach is successful used in others different applications on maintenance context (see [29,30,31,32]).

6 Conclusion

The purpose of this paper was to develop a multicriteria model to support decision-making in a combined burn-in-replacement policy. In comparison with classical models, a valuable enrichment of the results can be observed in favour of the multiple-criterion model proposed.

The multicriteria model was derived from the application of MAUT, which reflects the characteristics of the problem well and is a consistent method for cases in which there is uncertainty. By following the steps of MAUT, the decision-maker's preference structure can be obtained for both cost and reliability criteria, in addition to determining trade-offs between these two factors. As a result, the decision-maker can make a better choice, with the conviction that the decision reflects his/her wishes.

The illustration of an application of a decision model in the context of a packing supplier for the juice and milk industries was presented. The item subjected to burn-in is a component of the system provided by the supplier to the industries. This application highlighted the advantages provided by the model. Not only does this model provide a broader view of the problem, but it also meets the demand of the decision-maker to maintain high reliability in the field ($R_c(.)$) while investing the minimum cost ($C_{total}(.)$).

Acknowledgements. This study has been partially supported by CNPq (Brazilian Research Council).

References

1. Jiang, R., Jardine, A.K.S.: An optimal burn-in preventive-replacement model associated with a mixture distribution. Quality and Reliability Engineering International 23, 83–93 (2007)
2. Jiang, R., Murthy, D.N.P.: Mixtures of Weibull distributions-parametric characterization of failure rate function. Applied Stochastic Models and Data Analysis 14, 47–65 (1998)
3. Klutke, G.A., Kiessler, P.C., Wortman, M.A.: A Critical Look at the Bathtub Curve. IEEE Transactions on Reliability 52 (2003)
4. Almeida, A.T.: Multicriteria modelling of repair contract based on utility and ELECTRE I method with dependability and service quality criteria. Annals of Operations Research 138, 113–126 (2005)
5. Cha, J.H., Lee, S., Mi, J.: Bounding the optimal burn-in time for a system with two types of failure. Naval Research Logistics 51, 1090–1101 (2004)

6. Cha, J.H., Mi, J.: Bounds to optimal burn-in and optimal work size. Applied Stochastic Models in Business and Industry 21, 227–239 (2005)
7. Sheu, S.H., Chien, Y.H.: Minimizing Cost-Functions Related to Both Burn-In and Field-Operation Under a Generalized Model. IEEE Transactions on Reliability 53, 435–439 (2004)
8. Mi, J.: Age-smooth properties of mixture models. Statistics & Probability Letters 43, 225–236 (1999)
9. Drapella, A., Kosznik, S.A.: short communication combining preventive replacement and burn-in procedures. Quality and Reliability Engineering International 8, 423–427 (2002)
10. Finkelstein, M.S., Esaulova, V.: On an inverse problem in mixture failure rates modeling. Applied Stochastic Models in Business and Industry 17, 221–229 (2001)
11. Block, H.W., Savits, T.H., Wondmagegnehu, E.T.: Mixtures of distributions with increasing linear failure rate. Journal of Applied Probability 40, 485–504 (2003)
12. Abraham, B., Nair, N.U.: On Characterizing mixtures of some life distributions. Statistical papers 42, 387–393 (2001)
13. Barlow, R.E., Proschan, F.: Mathematical Theory of Reliability. Wiley, New York (1965)
14. Finkelstein, M.S., Esaulova, V.: Why the mixture failure rate decreases. Reliability Engineering and System Safety 71, 173–177 (2001)
15. Belton, V., Stewart, J.: Multiple criteria decision analysis - An integrated approach. Kluwer Academic Publishers, London (2002)
16. Figueira, J., Greco, S., Ehrgott, M.: Multiple criteria decision analysis: state of the art surveys. Springer, USA (2005)
17. Keeney, R.L., Raiffa, H.: Decisions with multiple objectives: preferences and value trade-offs. Cambridge University Press, United Kingdom (1993)
18. Fishburn, P.C.: Independence in utility with whole product sets. Operations Research 13, 28–45 (1965)
19. Raiffa, H.: Decision Analysis: Introductory Lectures on Choices Under Uncertainty. Hill College Custom Series (1997)
20. Keeney, R.: Value-focused Thinking: a path to creative decisionmaking. Harvard University Press, United States of America (1992)
21. Cha, J.H.: On a better Burn-in Procedure. J. Appl. Prob. 37, 1099–1103 (2000)
22. Kuo, W., Kuo, Y.: Facing the Headaches of Early Failures: A State-of-the-Art Review of Burn-In Decisions. Proceedings of The IEEE 71, 1257–1266 (1983)
23. Cha, J.H.: Stochastic Model for Burn-In Procedures in Accelerated Environment. Naval Research Logistics 53, 226–234 (2006)
24. Mi, J.: Burn-in and Maintenance policies. Adv. Appl. Prob. 26, 207–221 (1994)
25. Ferreira, R.J.P., de Almeida, A.T., Cavalcante, C.A.V.: A multi-criteria decision model to determine inspection intervals of condition monitoring based on delay time analysis. Reliability Engineering & Systems Safety 94, 905–912 (2009)
26. Cavalcante, C.A.V., de Almeida, A.T.: A multicriteria decision aiding model using PROMETHEE III for preventive maintenance planning under uncertain conditions. Journal of Quality in Maintenance Engineering 13, 385–397 (2007)
27. Scarf, P.A., Cavalcante, C.A.V., Dwight, R., Gordon, P.: An age-based inspection and replacement policy for heterogeneous components. IEEE Transactions on Reliability 58, 641–648 (2009)
28. Scarf, P.A., Cavalcante, C.A.V.: Hybrid block replacement and inspection policies for a multi-component system with heterogeneous component lives. European Journal of Operational Research 206, 384–394 (2010)

29. de Almeida, A.T.: Multicriteria decision making on maintenance: Spares and contracts planning. European Journal of Operational Research 129, 235–241 (2001)
30. Cavalcante, C.A.V., Ferreira, R.J.P., de Almeida, A.T.: A preventive maintenance decision model based on multicriteria method PROMETHEE II integrated with Bayesian approach. IMA Journal of Management Mathematics 21, 333–348 (2010)
31. de Almeida, A.T., Bohoris, G.A.: Decision theory in maintenance decision theory. Journal of Quality in Maintenance Engineering 1, 39–45 (1995)
32. Cavalcante, C.A.V., de Almeida, A.T.: A multicriteria decision aiding model using PROMETHEE III for preventive maintenance planning under uncertain conditions. J. Qual. Maint. Eng. 13, 385–397 (2007)

A Notation

Notation:

$f_i(t)$ pdf of components from sub-population i=1,2
p mixing parameter
η_i scale parameter of Weibull pdf of components from sub-population i=1,2
β_i shape parameter of Weibull pdf of components from sub-population i=1,2
$f_{mixed}(t)$ mixed probability density function (pdf)
$F_{mixed}(t)$ mixed cumulative distribution function (cdf)
$h_{mixed}(t)$ mixed failure rate function
b burn-in time
t age at preventive replacement during the wear-out phase
m number of repairs during the burn-in procedure
$E(m)$ expected number of repairs during the burn-in procedure
$E(t_b)$ expected burn-in time for a given item
C_r cost of repairs during the burn-in process
C_0 cost-rate of the burn-in process
C_p cost of preventive replacement
C_f cost of failure ($> C_p$)
$G(b, x)$ post-burn-in distributio
$R_c(b, x)$ post-burn-in reliability (conditional reliability function)
$C_b(b)$ expected burn-in cost
$C_{total}(b, y)$ expected total cost per operational time

Applying a Multicriteria Decision Model So as to Analyse the Consequences of Failures Observed in RCM Methodology

Marcelo Hazin Alencar and Adiel Teixeira Almeida

Department of Management Engineering, Federal University of Pernambuco,
PO Box 7462, Recife, PE, 50.722-970, Brazil

Abstract. Competitiveness among organizations, resulting from globalization, has seen to it that the introduction of or improvement in new processes and methodologies has substantially intensified in recent years. In this context, this paper puts forward a model application that incorporates a multicriteria decision approach to RCM (Reliability-centered Maintenance) methodology. In this model, a quantitative analysis of the consequences of failures is presented, which provides the maintenance manager with important data so that management decisions can be taken based on the decision maker's preferences. Thus, the model seeks to contribute to an approach that has become well-established within the maintenance area, since it provides more structured decision making, by using a quantitative multidimensional approach that takes into account the uncertainties associated with the problem.

Keywords: RCM (Reliability-Centered maintenance), Failure consequences, Multiattribute utility theory, Decision analysis.

1 Introduction

The dynamics of the global economy and its increasing complexity generate consequences for organizations [1]. This also applies to maintenance. In seeking performance levels that match the best world standards, most companies direct their efforts to ensuring that goals such as quality, productivity and reduced costs are achieved [2]. Checking on maintenance activities is a crucial part of these efforts. Effective and efficient maintenance extends the lifetime of equipment, improves availability rates and maintains facilities and components under appropriate conditions of use. However, it is worth keeping in mind that maintenance has not always been associated with reaching these goals. With respect to maintenance management it can be verified that the evolution of maintenance can be monitored over the past century, more specifically from the 1940s, and divided into three generations [3,4]:

- 1st generation - Covers the period up to Second World War, when industry was not highly mechanized. Maintenance was simple and there was no need for highly skilled maintenance operators;

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 594–607, 2011.

- 2nd generation - in the 1950's, due to increased mechanization, complex machines were introduced. The concept of preventive maintenance was introduced. Maintenance plans and control systems were developed for the control of costs associated with maintenance;
- 3rd generation - In this generation, in the mid to late 1970s, there were major changes associated with maintenance. These changes were related to new expectations, new research and new techniques. With respect to the new expectations, what stand out are the search for the increased availability and reliability of plant, increasing the level of security, improving product quality, environmental protection, increasing the life cycle of equipment and greater efficiency related to cost control. As for new research, changes were introduced in beliefs regarding the relationship between age and failure.

Despite the remarkable evolution of maintenance over the years, the classical problems of maintenance in modern industries became basically ones of insufficient proactive maintenance, repetition of frequently-occurring problems, erroneous maintenance work, unnecessary preventive maintenance, incomplete preventive maintenance actions and also the lack of predictive maintenance applications [3]. For these reasons, there was a need to seek appropriate methodologies so as to develop strategies and programmatic approaches to dealing with such problems. Added to this scenario, there was an increase in requirements regarding availability, reliability and life cycle and an increasing awareness in society of the need to preserve the environment and ensure user safety with respect to industrial processes and products. This situation created conditions which led to the emergence of the Reliability-Centered Maintenance (RCM) methodology. RCM offers a better systematic and a more efficient programmatic approach to optimising plant and equipment maintenance [3]. However, this paper will focus on a specific point within the RCM approach: the consequences of failures as revealed by a multicriteria approach.

2 RCM - Reliability Centered Maintenance

The first industry systematically to confront challenges associated with choices related to maintenance was the commercial aviation industry [4]. From this confrontation, the first event related to the origin of this RCM is recorded namely the need for certification of a line of aircraft. This resulted in the development of decision-making processes known in aviation as MSG3, and, in other industries, as RCM (Reliability-Centered Maintenance).

RCM is a systematic methodology that seeks to allocate effective predictive and preventive maintenance, helping to prevent the dominant causes of failure of critical equipment. Moreover it enables adequate levels of component availability and low cost to be achieved [5].

The central goal of the RCM is to determine the actions necessary to ensure that all physical assets continue to perform their functions within their operating environment [6].

RCM identifies ways in which the system can fail. RCM should be applied by managers who wish to attain high standards of maintenance for the facilities [7,8].

RCM focuses on maintaining the system rather than on the operation of the equipment [9]. Four terms are significant in this approach: system, subsystem, functional failure, and failure mode.

These terms are also called RCM functional components. In this context, RCM requires five steps to be followed [4]:

– Define the functions of each physical asset within the operating context, together with the associated desired standards of performance.
– Identify failures that may occur in the physical asset. In RCM this is conducted at two levels: identifying the circumstances that cause a functional failure and investigating the events that can cause a functional failure of the physical asset.
– Check the events that probably cause each functional failure. These events are known as failure modes. In this step, it is also important to identify the cause of each failure in a very detailed way in order to ensure time and effort is not wasted by addressing symptoms rather than causes.
– The fourth step involves listing the effects of failure, namely describing what happens when each failure mode occurs.
– Finally, the last step is to identify and analyze the consequences of failure.

3 Failure Consequences

RCM classifies these consequences into four groups [4,10,11]:

– Hidden consequences of failure: They do not have a direct impact, but expose the organization to multiple failures with serious, often catastrophic, consequences.
– Safety and environmental consequences: A failure has safety consequences related to the possibility of death or injury. The environmental consequences may mean that a company has violated a regional, national or even international environmental standard.
– Operational consequences: A failure has operational consequences if these affect production.
– Non-operational consequences: The evident failures which are in this category did not affect either safety or production, and involve only the direct cost of repair.

Depending on the facilities, a failure could generate insignificant or negligible consequences or affect systems vital to the company and society or the safety of human beings. In RCM, the consequences are analyzed by studying the impacts of the effects of failure modes on system operations, the environment, physical security and the economy of the process. In this context, this paper focuses mainly on the construction of a multicriteria decision model for assessing the consequences of failures.

4 Multicriteria Decision Aid

Multicriteria Decision Aid, in principle, introduces and establishes a relationship of (subjective) preferences among the alternatives that are being considered under the influence of multiple criteria in decision making. It is based on the analysis of decision problems where, for the actors, there are conflicting criteria in decision making. In summary, multiple criteria decision aid consists of a set of methods and techniques to assist or support individuals and organizations to make decisions under the influence of a multiplicity of criteria [12].

From an operational point of view, the methods of multicriteria decision aid are distinguished by their ability to consider many different points of view, characterized as conflicting with each other, thus enabling an integrated assessment of the problem [13].

In this context, the decision process has become an object of study on the world stage due to the great importance attached to consequences.

Emergency situations, whether caused by humans or by nature, require coherent and effective management involving complex decisions. Many conflicting objectives must be resolved and priorities need to be defined, while the many perspectives of various stakeholders should converge towards a consensus [14]. Thus, Multicriteria decision analysis (MCDA) can help this process of decision making, especially in a context that involves many uncertainties and conflicting objectives, as can be observed in this paper.

With respect to maintenance, the determination of maintenance strategy for each component in a plant is very complex due to the difficulties in acquiring data, the number of factors that must be taken into account (investment required, safety issues and the environmental costs of failures, reliability of policies, mean time between failures, mean time to repair, etc.), their subjectivities, the large number of machines to be considered within a plant, and moreover, in some cases, there is the fact that the plant may not yet have been set up [15]. To solve this problem, some multicriteria decision approaches (MCDM - Multi-Criteria Decision Making) help the decision maker to decide what maintenance strategy to adopt. In this context, the use of MAUT (multiattribute utility theory) is singled out, and this is summarized below.

4.1 MAUT (Multiattribute Utility Theory)

MAUT, a compensatory method of decision support, can be used to aggregate value preferences and consequences with respect to multiple dimensions, taking into account the decision maker's preferences and behavior, as it considers cases with uncertainty in a clear measure of risk [12,16]. In addition, MAUT includes utility theory axioms. The basic idea of utility theory is to quantify the decision maker's wishes, and lists the assets with the values that represent a rule of choice for the decision maker.

The process of eliciting the utility function involves understanding and modeling the structure of the decision maker's preferences with respect to the consequences [17].

MAUT takes into consideration the decision-maker's preferences in the form of a utility function that is defined by a set of attributes [18]. MAUT provides a logical way to deal with tradeoffs between conflicting goals [19].

In the same way, if an appropriate utility is assigned to each possible outcome and the expected utility of each alternative is calculated, the best course of action is held to be the alternative with the highest expected utility [19].

The utility function represents an indicator that combines the consequence dimensions in a desirability index. In the context of MAUT, the multi-attribute utility function must be estimated according to the domain of the consequences This function is estimated by using a structured protocol based on the axiomatic structure of utility theory. This theory allows the probabilistic assessment of the consequences under uncertainty [20].

The process of assessing a multi-attribute utility function consists of five steps: introducing the terminology and idea; identifying relevant hypotheses of independence; assessing the conditional utility function; evaluating the scale constants; and checking consistency [19].

5 Multicriteria Decision Model for Assessing the Consequences of Functional Failures

The choice of a multicriteria decision support depends on several factors. Terms that should be highlighted are: the problem analyzed, the context considered, the availability of information and its degree of accuracy, the rationality required, the decision makers preference structure and the problematic [12]. In this context, the most appropriate definition of rationality for the decision maker in the problem considered is an issue of great importance, since this will involve a compensatory or non-compensatory approach. Therefore, the decision model proposed in this paper sets out to improve the RCM approach by incorporating contributions from MAUT [12].

The choice of MAUT is due to the fact that it presents a well-structured protocol, supported because it has a very solid and consistent axiomatic structure, involving multiple criteria, for decision-makers. Moreover, at the stage of probabilistic modeling, uncertainties are inserted into the axiomatic structure allowing a more consistent approach with respect to applying MAUT to multicriteria decision problems under uncertainty [12]. This stage of probabilistic modeling complements that of modeling decision-makers's preferences.

MAUT is applied in order to make a quantitative analysis of the consequences of failures, taking into account decision-makers' preferences and value judgments. By applying this model, the decision maker is faced with more significant results that can be used as input in the process of maintenance management.

The new steps to be introduced in Reliability Centered Maintenance are shown below:

- Identifying the dimensions of consequences
- Analysing consequences

- Probabilistic modeling
- Defining the global utility index for each item considered
- Ranking the alternatives

Therefore, first there is a need to compare the classical RCM approach with the RCM approach to incorporating the multicriteria model. Comparing Figs. 1 and 2 is a concise way of understanding the steps of both.

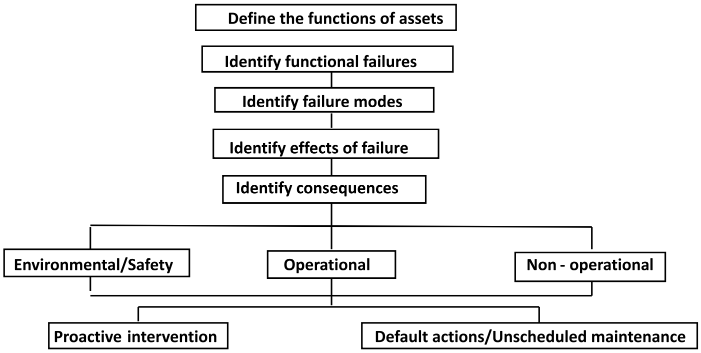

Define the functions of assets
Identify functional failures
Identify failure modes
Identify effects of failure
Identify consequences
Environmental/Safety
Operational
Non - operational
Proactive intervention
Default actions/Unscheduled maintenance

Fig. 1. RCM process (adapted from [4,10])

Fig. 2 depicts the inclusion of the multicriteria model proposed in the RCM process. Some steps of the RCM are maintained such as define the functions of assets, identify functional failures, identify failure modes and identify the effects of failure.

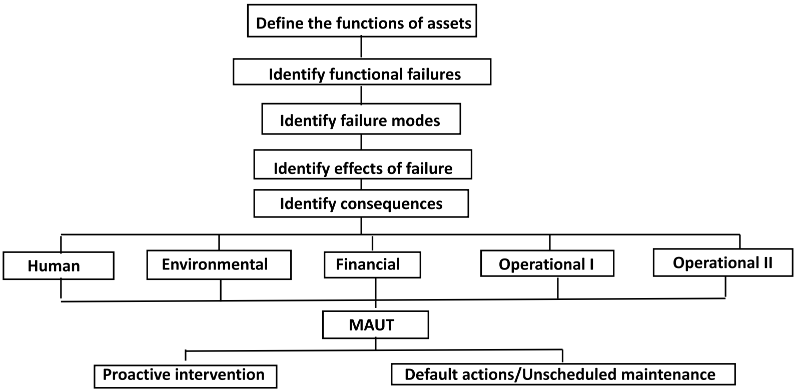

Define the functions of assets
Identify functional failures
Identify failure modes
Identify effects of failure
Identify consequences
Human
Environmental
Financial
Operational I
Operational II
MAUT
Proactive intervention
Default actions/Unscheduled maintenance

Fig. 2. RCM with the inclusion of the proposed model

5.1 Identifying the Dimensions of the Consequences

The consequences of failures are evaluated in the model proposed in five groups defined as the dimensions of the consequences, some of the characteristics differing from those established by the RCM approach:

- Human dimension (h) - In this dimension, the damage with respect to people affected by the consequences of failures is considered.
- Environmental dimension (e) - This takes into account the area affected as a result of the failure.
- Financial dimension (f) - This takes into account the financial losses with regard to the consequences of failures.
- Operational I dimension (o') - The influence of the consequences of failures on the behavior of the production system is considered. The consequences of failure affect the production process without interrupting the operation.
- Operational II dimension (o") - The influence of the consequences of failure on the behavior of the production system is considered in this dimension. But, in this case, the consequence of failure completely interrupts the operation.

5.2 Analysing the Consequences

In order to analyze the consequences, elements of decision theory will be used in which θ is defined as the state of nature expressing the uncertainty associated with the problem. θ represents scenarios resulting from failure modes in equipment/ the system. The consequences are represented by c and the set of all possible actions under consideration is represented by A.

A probabilistic approach is presented to deal with the associated uncertainties in A using a probability distribution over consequences and by eliciting utility functions for these consequences. The probability of each state of nature is defined as $\pi(\theta) \cdot U(\theta, a_i)$ is the utility when scenario θ and action a_i are considered [21].

The utility is calculated by combining the probability of the consequences c in A (deemed the consequence function, $P(c|\theta, a_i)$.

The values of the utility functions are defined in an interval scale between the extremes [0, 1]. "0" is related to the "least preferred" while the extreme "1" is related to the "most preferred" [19].

The utility function of these consequences is shown by Eq. (1) for discrete cases:

$$U(\theta, a_i) = \sum_c P(c|\theta, a_i) U(c) \ . \tag{1}$$

Finally, Eq. (2) shows the utility function of these consequences for continuous cases:

$$U(\theta, a_i) = \int_c P(c|\theta, a_i) U(c) dc \ . \tag{2}$$

when the scenario θ and the action a are considered. Utility combines the consequence function and utility function of these consequences.

5.3 Probabilistic Modeling

The objective of this model is to obtain a utility function U of an individual or group of individuals whose values are of interest. The choice of MAUT as a method to be applied is equivalent to choosing a type of compensation among criteria. The elicitation of utility functions occurs in an interval of closed consequences, where a nil result (no impact) is associated with maximum utility, while the minimum utility value is related to the largest of the consequences estimated [22].

The approach considered consists of subdividing the assessment of U into parts, working with these parts and then aggregating them later [23]. This requires the decision-maker's final qualitative judgment values to be affirmed and quantified.

It is important to mention that in intracriteria evaluation, it is necessary to define the functions of the consequences, because the utility combines the consequence function and the utility function of these consequences. Thus, different considerations are established for defining the consequence function in each dimension.

After having obtained the values of U and k, using elicitation procedures based on the comparison lottery [19], the multi-attribute utility function is then calculated.

5.4 Defining the Overall Utility Index

In this model the additive utility function was established. This function was obtained by using unidimensional utilities represented by Eq. (3):

$$U(h,e,f,o',o'') = k_1U(h) + k_2U(e) + k_3U(f) + k_4U(o') + k_5U(o'') \ . \tag{3}$$

Where:

- k_i is a scale constant that represents the value of the tradeoff;
- The sum of k's needs to be equal to "1" $(k_1 + k_2 + k_3 + k_4 + k_5 = 1)$;
- These scale constants are elicited by procedures based on comparing lotteries described in [19].

5.5 Ranking the Alternatives

Another form of information can be analyzed in utility theory: the ranking of multi-attribute utility values. This comes about due to the axiomatic structure of utility theory.

The interval scale of utility function allows the incremental value to be compared to the failure modes [19]. Using the interval scale, it may be affirmed that the difference $U(MF_x)_{\beta x} - U(MF_y)_{\beta x+1}$ is M times greater than the difference $U(MF_y)_{\beta x+1} - U(MF_z)_{\beta x+2}$. This can be seen from the increment ratio (IR) of these differences since $IR = (U(MF_x)_{\beta x} - U(MF_y)_{\beta x+1})/(U(MF_y)_{\beta x+1} - U(MF_z)_{\beta x+2})$.

6 Numerical Application

In order to illustrate the model proposed for evaluating the consequences of failures obtained from the RCM approach, a numerical application based on a case study is presented in this section, which considers the 16 most relevant components for the system, as well as there being a single user of the system.

Human (h), environmental (e), financial (f), operational I (o') and operational II (o") dimensions were considered to estimate the consequences.

For each component x considered in the system, a failure mode FM_x was adopted with an occurrence probability $\pi(\theta)_x$ as can be seen from Table 1.

Table 1. A prior probability of failure modes for each component

Component x	Failure Mode FM_x	$\pi(\theta)_x$
1	FM_1	0.0766
2	FM_2	0.0256
3	FM_3	0.0578
4	FM_4	0.0333
5	FM_5	0.0835
6	FM_6	0.0259
7	FM_7	0.0768
8	FM_8	0.0493
9	FM_9	0.0876
10	FM_{10}	0.0087
11	FM_{11}	0.07
12	FM_{12}	0.0563
13	FM_{13}	0.0367
14	FM_{14}	0.0154
15	FM_{15}	0.0958
16	FM_{16}	0.0757

With the support of computational tools, the probabilities of the consequences were estimated for the dimensions of the consequences. Similarly, the one-dimensional utility functions were obtained using one of the equations described in subsection 5.2.

Following the elicitation procedure based on the comparison of lotteries [19], the scale constants are established: $k_1 = 0.19, k_2 = 0.13, k_3 = 0.27, k_4 = 0.11, k_5 = 0.30$. Using these values, the multi-attribute utility function U(h, e, f, o', o") is elicited from the decision-maker, following structured protocols that consider the choice between the consequences of failure functions and lotteries with specific probabilities between the best and worst cases [19].

A conversion scale is introduced into the values of the multi-attribute utility function so as to facilitate the interpretation of these values, as can be seen in Table 2.

The ranking of the multiattribute utility for each failure mode considered is given in Table 3.

Table 2. Failure modes and multiattribute utility values

Failure Mode FM_x	U(h,e,f,o',o")	Conversion scale
FM_{01}	0.95595	0.25952
FM_{02}	0.98427	0.80575
FM_{03}	0.95809	0.30082
FM_{04}	0.97875	0.69934
FM_{05}	0.96341	0.40343
FM_{06}	0.99004	0.91700
FM_{07}	0.96705	0.47368
FM_{08}	0.97736	0.67248
FM_{09}	0.94705	0.08788
FM_{10}	0.99434	1
FM_{11}	0.97777	0.68042
FM_{12}	0.98636	0.84600
FM_{13}	0.97965	0.71658
FM_{14}	0.97131	0.55575
FM_{15}	0.94250	0
FM_{16}	0.98846	0.88658

Table 3. Ranking of the multiattribute utility for each failure mode

Ranking position (βx)	Failure Mode FM_x	$U(FM_x)_{\beta x}$
β_{01}	FM_{15}	0
β_{02}	FM_{09}	0.08788
β_{03}	FM_{01}	0.25952
β_{04}	FM_{03}	0.30082
β_{05}	FM_{05}	0.40343
β_{06}	FM_{07}	0.47368
β_{07}	FM_{14}	0.55575
β_{08}	FM_{08}	0.67248
β_{09}	FM_{11}	0.68042
β_{10}	FM_{04}	0.69934
β_{11}	FM_{13}	0.71658
β_{12}	FM_{02}	0.80575
β_{13}	FM_{12}	0.84600
β_{14}	FM_{16}	0.88658
β_{15}	FM_{06}	0.91700
β_{16}	FM_{10}	1

The interval scale of the utility function allows comparison of the increments of utility with respect to failure modes [19]. These increments are presented in Table 4.

The values given in Table 4 provide important information for the company, since they serve as input data for management and maintenance planning, thus aiding the process for allocating resources. Examples of information that can be found are: the difference between the values of the utilities associated with

Table 4. Comparisons of utility values and ratios of the increments of utility among prioritized failure modes

Ranking (βx) position	Failure Mode FM_i	$U(FM_x)_{\beta x}$ $-U(FM_y)_{\beta x+1}$	Increment Ratio IR
β_{01}	FM_{15}	0.08788	0.51200
β_{02}	FM_{09}	0.17164	4.15593
β_{03}	FM_{01}	0.0413	0.40249
β_{04}	FM_{03}	0.10261	1.46064
β_{05}	FM_{05}	0.07025	0.85598
β_{06}	FM_{07}	0.08207	0.70308
β_{07}	FM_{14}	0.11673	14.70151
β_{08}	FM_{08}	0.00794	0.41966
β_{09}	FM_{11}	0.01892	1.09745
β_{10}	FM_{04}	0.01724	0.19334
β_{11}	FM_{13}	0.08917	2.21540
β_{12}	FM_{02}	0.04025	0.99187
β_{13}	FM_{12}	0.04058	1.33399
β_{14}	FM_{16}	0.03042	0.36651
β_{15}	FM_{06}	0.083	-
β_{16}	FM_{10}	-	-

the failure modes FM_{14} and FM_{08} is 0.11673; and the difference between the values of the utilities associated with the failure modes FM_{08} and FM_{11} is 0.00794. This means that an increment in the utility values from FM_{14} to FM_8 is approximately 15 times greater than the increment in the utility values from FM_{08} to FM_{11}. This analysis is based on the interval scale provided by the utility function [16].

A sensitivity analysis was conducted based on the data and parameters analyzed to verify the robustness of the model. Results show that the procedure was robust.

7 Results

An important point to note is that the results presented here can be correlated or integrated with the results of other studies presented in the literature, as demonstrated below.

A combination between decision theory and influence diagrams is proposed [24], in order to present the relationships between decisions, random quantities and preferences. It uses performance indicators that show whether one or several goals are achieved. These indicators are combined into a value function or utility function, with a view to establishing a set of performance indicators for each objective and thus obtaining an optimal maintenance model. With a focus that is different from the model proposed in [24], this paper also targets the construction of a multicriteria decision model to evaluate the consequences of failures. It proposes that the consequences should be restructured into five dimensions, unlike the approach proposed by the RCM. Thus, the consequences of the failures can be evaluated quantitatively. Another important point in this paper is the ranking of the multiattribute utility values obtained.

An evaluation system to aid decision making regarding the feasibility of developing predictive maintenance programs is presented [25], using a combination of operational research tools such as the Analytical Hierarchy Process, decision rules and Bayesian tools. The company's ability to implement a predictive maintenance program as well as to assess the need to implement this program are evaluated, taking into account decision variables such as safety, quality, plant availability, repair costs, etc. Thus, the model proposed in this present paper can add to the contribution in [25] helping to choose the best maintenance policy, since it incorporates some improvements made to RCM methodology.

In [17,26] basic elements of decision theory that are used in several studies in the literature are presented, including in this paper: a set of possible actions, the consequences, the utility function, multiattribute utility theory (MAUT), elicitation and checking of consistency, optimization through the Bayesian approach and sensitivity analysis. The decision model [17], as well as its step-by-step approach served as the basis for developing this multicriteria decision model to evaluate the consequences of failures uncovered when using RCM methodology.

The results from the ranking of the multi-attribute utility of the consequences for failure modes considered in the studies undertaken provide managers with input data to support decision making associated with maintenance planning and maintenance management, thus allowing integration with the results from the model [27], since this proposes a multicriteria analysis linked to maintenance planning, specifically in providing spare parts and selecting repair contracts. Similarly, this integration can also be checked in [28], which addresses the issue of hiring outside companies to carry out repairs and maintenance on their systems. As to the possible integration of the results of this paper in [27] as well as the possible integration in [28], it is observed that, by ranking the utilities of the consequences associated with each failure mode, it is possible to prioritize possible actions to be taken, and also to define the responsibility for repairs of parts or systems (whether or not these are to be conducted by outside companies).

8 Conclusions

Currently, the RCM approach is widely used to ensure that assets continue to perform their functions within the operating environment. The use of a multicriteria decision model to evaluate the consequences obtained from an RCM approach is proposed in order to ensure an improvement in this process.

The multicriteria decision model makes a quantitative evaluation of the consequences of failure obtained from the RCM approach, thus allowing managers to maintain data that support decision making and serve as input for the process of maintenance management.

The utility function is obtained for each of the failure modes considered, taking into account consequence dimensions. The decision-makers' preferences and value judgments are considered.

In the same way, consequence groups established under the RCM approach were restructured into new five dimensions. The consequences that affect safety and the environment were separated into two new dimensions in order to generate more accurate assessments, given that environmental questions and human lives are highly valued in society today.

In conclusion, the improvements obtained by implementing a multi-criteria decision model seeks to generate consistent results that can assist managers when they draw up maintenance plans. Furthermore, it is worth emphasizing the advantage of using the utility function in a context, such as maintenance, where the criteria are related to probability distributions. This has also been observed in previous studies [26,27,28].

Acknowledgements. This study was partially supported by CNPq (the Brazilian Research Funding Bureau) and TERMOPE (P&D TPE 29) for which the authors are grateful.

References

1. Bourne, L., Walker, D.H.T.: The paradox of project control. Team Performance Management 11, 157–178 (2005)
2. Swanson, L.: Linking maintenance strategies to performance. Int. J. of Prod. Economics. 70, 237–244 (2001)
3. Deshpande, V.S., Modak, J.P.: Application of RCM to a medium scale industry. Reliability Engineering and System Safety 77, 31–43 (2002)
4. Moubray, J.: Reliability-centered Maintenance. Industrial Press Inc., North Carolina (1997)
5. Martorell, S., Sanchez, A., Munoz, A., Pitarch, J.L., Serradell, V., Roldan, J.: The use of maintenance indicators to evaluate the effects of maintenance programs on NPP performance and safety. Reliability Engineering and System Safety 65, 85–94 (1999)
6. Fonseca, D.J., Knapp, G.M.: An expert system for reliability centered maintenance in the chemical industry. Expert Systems with Applications 19, 45–57 (2000)
7. Marquez, F.P.G., Schmidt, F., Collado, J.C.: A reliability centered approach to remote condition monitoring: a railway points case study. Reliability Engineering and System Safety 80, 33–40 (2003)
8. Abdul-nour, G., Beaudoin, H., Ouellet, P., Rochette, R., Lambert, S.: A Reliability based maintenance policy: a case study. Computers Ind. Eng. 35, 591–594 (1998)
9. Deshpande, V.S., Modak, J.P.: Application of RCM for safety considerations in a steel plant. Reliability Engineering and System Safety 78, 325–334 (2002)
10. Hipkin, L., Lockett, A.G.: A study of maintenance technology implementation. Omega 23, 79–88 (1995)
11. Carretero, J., Prez, J.M., Garcia-Carballeira, F., Caldern, A., Fernndez, J., Garca, J.D., Lozano, A., Cardona, L., Cotaina, N., Prete, P.: Applying RCM in large scale systems: a case study with railway networks. Reliability Engineering and System Safety 82, 257–273 (2003)

12. Almeida, A.T.: Knowledge and Use of Methods for Multicriteria Decision Making (Text in Portuguese: O Conhecimento e o Uso de Mtodos Multicritrio de Apoio a Deciso). Editora Universitria da UFPE, Recife (2010)
13. Munda, G.: Social Multi-criteria evaluation for a sustainable economy. Springer, Berlin (2008)
14. Gelderman, J.: Multi-criteria decision support and evaluation of strategies for nuclear remediation management. Omega 37, 238–251 (2009)
15. Bevilacqua, M., Braglia, M.: The analytic hierarchy process applied to maintenance strategy selection. Reliability Engineering and System Safety 70, 71–83 (2000)
16. Brito, A.J., Almeida, A.T.: Multi-attribute risk assessment for risk ranking of natural gas pipelines. Reliability Engineering and System Safety 94, 187–198 (2009)
17. Almeida, A.T., Bohoris, G.A.: Decision theory in maintenance decision making. Journal of Quality in Maintenance Engineering 1, 39–45 (1995)
18. Afgan, N.H., Carvalho, M.G., Pilavachi, P.T., Martins, D.: Evaluation of natural gas supply options for Southeast and Central Europe: Part 2-Multi-criteria assessment. Energy Conversion and Management 49, 2345–2353 (2008)
19. Keeney, R.L., Raiffa, H.: Decisions with multiple objectives: preferences and value trade-offs. John Wiley & Sons, New York (1976)
20. Ferreira, R.J.P., Almeida, A.T., Cavalcante, C.A.V.: A multi-criteria decision model to determine inspection intervals of condition monitoring based on delay time analysis. Reliability Engineering and System Safety 94, 905–912 (2009)
21. Berger, J.O.: Statistical decision theory and Bayesian analysis. Springer, New York (1985)
22. Vincke, P.: Multicriteria Decision-aid. John Wiley & Sons, Brussels (1992)
23. Keeney, R.: Value-focused thinking: a path to creative decision making. Harvard University Press, London (1992)
24. Vatn, J.: Maintenance Optimisation From a Decision Theoretical Point of View. Reliability Engineering and System Safety 58, 119–126 (1997)
25. Carnero, M.: An Evaluation System of the Setting Up of Predictive Maintenance Programmes. Reliability Engineering and System Safety 91, 945–963 (2006)
26. Almeida, A.T., Souza, F.M.C.: Decision Theory In Maintenance Strategy For A Two-Unit Redundant Standby System. IEEE Transactions on Reliability 42, 401–407 (1993)
27. Almeida, A.T.: Multicriteria Decision Making on Maintenance: Spares and Contracts Planning. Eur. J. of Oper. Res. 129, 235–241 (2001)
28. Almeida, A.T.: Multicriteria Modelling of Repair Contract Based on Utility and ELECTRE I Method with Dependability and Service Quality Criteria. Annals of Oper. Res. 138, 113–126 (2005)

Supplier Selection Based on the PROMETHEE VI Multicriteria Method

Luciana Alencar and Adiel Almeida

Department of Management Engineering, Federal University of Pernambuco,
PO Box 7462, Recife, PE, 50.722-970, Brazil

Abstract. Nowadays, supplier selection has become a strategic activity for many companies that wish to increase the competitiveness of their supply chain. The tendency is to maintain a long-term relationship based on trust and commitment. To achieve this aim, it is necessary to re-structure the supplier selection process, incorporate more criteria in the evaluation and deal in an appropriate way with the preferences of the decision-makers (DMs) involved. Thus, this paper presents a structure for selecting suppliers that considers the preferences of multiple DMs involved in the process. It is considered that their preferences do not present great variation, it therefore being possible to deal with them adequately and to select the supplier who presents the best compromise using the PROMETHEE VI method. A numerical application is presented.

Keywords: supplier selection, group decision, multicriteria methods.

1 Introduction

Supply chain management (SCM) is a set of management processes for planning, implementing and controlling operations from the ultimate supplier to the ultimate customer, with strategic orientation towards cooperative efforts to synchronize and dovetail intra-firm and inter-firm relationships. One of the processes of SCM is supplier relationship management (SRM). This involves the way in which the company interacts with its suppliers. According to [1] SRM is becoming more and more critical for companies to the extent that they concentrate on core competencies and rely on suppliers to have more competitive advantage in their processes.

Companies must know each type of product or service to be outsourced so as to decide on the type of relationship that will be undertaken with their suppliers. Relationships vary from arm's length to high involvement and can be short-term, mid-term or long-term. Therefore, supplier selection is a decisive process for choosing the supplier who is most adequate for the needs and objectives of the company. Increasingly, the competence of a company in meeting its client's needs is dependent on the quality of the products supplied and services offered by outsourced companies. This fact shows the importance of having suppliers who are qualified and able to meet their contractors strategic objectives. Thus, the

R.H.C. Takahashi et al. (Eds.): EMO 2011, LNCS 6576, pp. 608–618, 2011.

selection process can no longer be based only on choosing those offer low costs but rather on those factors that are most in line with the company's wishes. What has become required is a higher qualifier level and for many criteria to be considered at selection, such as flexibility, willingness to cooperate, organizational culture, credibility, and so forth.

The supplier selection process is of the utmost importance in effectively managing current supply chain networks as it is essential to achieve customer satisfaction [2]. A study identified 23 criteria considered by managers to proceed with the selection process under different scenarios [3]. Since then, many studies were developed using a multicriteria approach. Thus, a supplier selection decision is regarded as a multicriteria decision problem. The problem can become more and more complex depending on the number of criteria involved, the number of decision-makers (DM's) involved in the process and the type of criteria (qualitative and/or quantitative) considered. Many studies were developed for supplier selection using multicriteria decision aid methods [4,5,6,7,8,9,10].

This paper addresses the supplier selection process when there is more than one decision-maker involved in the decision process, but their preferences show little variation from each other. A framework for supplier selection will be presented in the context of using the PROMETHEE multicriteria method to deal with the DM's preferences.

The paper is organized as follows: Sect. 2 presents a brief literature review regarding Supplier Selection; Sect. 3 presents a background in group decision and multicriteria methods; Sect. 4 describes the framework for supplier selection for a group of DM's; Sect. 5 gives a numerical application of the framework; Sect. 6 presents the concluding remarks of the study.

2 Supplier Selection

Supplier selection is a process in which suppliers are inspected, evaluated and chosen to become part of the supply network of an organization [11].

Many approaches were proposed for supplier selection, involving Analytic Hierarchy Process (AHP), Analytic Network Process (ANP), Data envelopment analysis (DEA), Fuzzy Logic, Genetic Algorithms (GA), Simple Multi-attribute Rating Technique (SMART), and so on [12]. Besides these, there are some hybrid approaches that combine different techniques to solve the problem [13,15]. Some approaches found in the literature are classified as [14]: categorical methods, method of linear weighted sum, method of distribution of costs, dimensional analysis, and evaluation of suppliers with a multicriteria method.

Some important aspects that should be taken into account when determining the criteria are presented [16], since they are essential for measuring the suitability of the supplier for the contractor's needs:

- Financial: The financial strength of the supplier is a firm guarantee for the stability of that supplier in the long term;
- Managerial Aspects: The organization and the supplier must have compatible approaches to management, especially when seeking integrated and strategic relationships;

- Technical: involve the provision of products and services of high quality;
- Support resources;
- Quality systems and processes;
- Globalization and localization.

Therefore, what is evident from the above list is the multicriteria character of the supplier selection problem in modern supply chains. In many cases, the decision to select these suppliers is taken by more than one DM. Thus, the use of techniques to support group decision is stressed.

The process of supplier selection involves analysis of various criteria, some quantitative and others qualitative. These decisions are becoming increasingly complex given the increase in outsourcing and business electronics [17]. As more and more experts (and conflicting opinions) are involved, the decision to select supplier becomes more complex and a problem of group decision.

There are several studies that propose methods for aggregating different opinions in group decision. One of the key issues is how to choose the function and aggregation operators to combine the different views of DM's in a single numerical value [17].

According to [18], in essence, the process of supplier selection is a problem of group decision-making under multiple criteria. The number of DMs involved is one factor that should be taken into account, together with the degree of uncertainty involved and the nature of the criteria when looking for the solution to such problems.

Some studies for supplier selection for a project taking the group decision approach into consideration were found [19,20,21].

3 Group Decision and Multicriteria Method

Group decision making involves many aspects that are intrinsic to each decision maker's (DM) individuality and nature. Multiple attribute group decision analysis (MAGDA) problems are defined as decision situations where a group of DMs express their preference on multiple criteria for a problem to be solved and try to find a common solution [22]. When a team of experts takes part in the decision process, it is expected that their opinions will differ, since each member of the group has different information at hand and only partially shares the goals of other members [23].

Two approaches are normally considered with multicriteria decision aid methods to consider/aggregate the group preferences [24]. In the first one, the DM's must agree in relation to the alternatives, criteria, performances, weights, thresholds and others parameters that are needed to reach a solution according to the problematic chosen. In the second, the DMs might change opinions and relevant information but each member defines his/her own criteria, their evaluations and the parameters necessary according to the multicriteria method that was chosen. After that, each DM is considered as a separate criterion and the information contained in his/her individual ranking is aggregated in a collective final ranking, using the same or another multicriteria method. This last approach is the one used in this study.

The multicriteria methodology area has a large set of tools the purpose of which is to aid the decision maker during the development of a decision process [25]. Among other things, this kind of decision making problem is characterised by the existence of several criteria of different natures with different value scales, quite often conflicting ones.

In this study, the method applied was PROMETHEE II and VI, which is a subdivision of the PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluation) method [26]. As it is possible to incorporate in PROMETHEE VI method the range of variation of weights [27], this method was chosen to treat the problem of a group of DMs with little divergence among their preferences regarding to the criteria weights. These divergences are incorporated in the range of weights variation.

The advantage of this method is that it requires additional, very clear information that can be obtained easily and managed well by both the decision maker and the analyst [28]. Besides, the method presents greater flexibility, since the selection of the type of preference function constitutes an important degree of freedom offered to the decision maker. He/she can have two degrees of freedom: the first is relative to the type of criteria generalized (six types) and the second to the thresholds to be defined.

In order to apply the PROMÉTHÉE VI method, it is necessary to use the PROMÉTHÉE II method and the Geometrical Analysis for Interactive Aid (GAIA) procedure [29].

4 Framework for Supplier Selection with Multicriteria Model

In this section the framework for supplier selection is presented using the PROMETHEE VI method, applied to situations in which there is a group of decision makers that will be responsible for supplier selection, but there is little divergence among their preferences [30]. In this situation, they are able to define a variation range (an upper level and a lower level) for the weight of each criterion.

The framework begins with identifying the decision criteria to be used in the selection process. Two types of criteria must be identified: the qualifiers and the bid winners' criteria. As to the qualifiers criteria, the evaluation of the alternatives must reach a required performance level to be, at least, considered in the process. The bid winners' criteria are those which exert influence on the company's competitive potential. The former are used in the filter stage and the latter in the final selection stage.

Thus, for those criteria that will be used in the final selection, it is necessary to establish the relative importance among them. As this structure is for a group of DMs with little divergence between their preferences, it is possible that they are not able to establish jointly a fixed value for each weight but they are able to consider an interval, considering all DMs' preferences. Thus, the PROMETHEE VI method is appropriate for this situation since it was developed for those cases in which the DM is not able to provide precise values for criteria weights. The intervals are fixed from a known value w_j, accepting a percentage Θ_j of variation of this value:

$$w_j \pm \Theta_j.w_j, \quad j = 1, 2, ..., k \ . \tag{1}$$

After establishing this information, the suppliers could be invited to participate in a briefing meeting. The suppliers who attend this meeting must deliver all the documentation needed to begin the selection process, in accordance with the information given by the focus company. In this meeting they will be informed about all the stages of the selection process.

The next stage is the filter. The potential suppliers will be analyzed regarding the qualifying criteria, defined previously. The companies that reach the minimum level required go forward to the next stage: the final selection, where bid winners criteria are used. After the criteria evaluation, the multicriteria method is applied. For the PROMETHEE VI application, the alternative net flows first need to be calculated by PROMETHEE II. With these flows, the projections of the alternatives such as the projections of the criteria in the GAIA plane could be undertaken. Through PROMETHEE VI, the variation space of criteria weights established by DMs is considered. The projection of this area in the GAIA plane is the DMs' freedom space. Therefore, any alternative positioned in this direction presents a good compromise with the DMs' preferences.

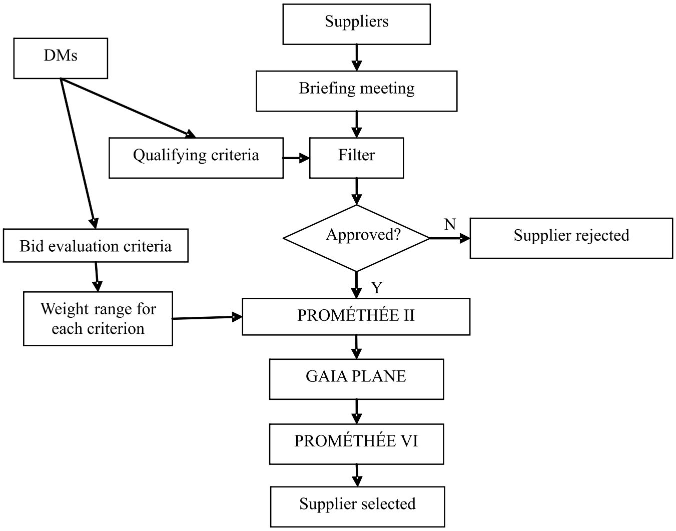


Fig. 1. Supplier selection framework

5 Numerical Application

In this section, what is set out is an application of the framework described in the last section for selecting consultants to carry out the designs for a construction project of a private telecommunication company [30].

The company's DMs are dissatisfied with the selection process that is used. A recent situation that occurred was a factory was built, where the supplier was selected based on lowest price, and, at the moment when he was to work on the special facilities required, he did not have the skills to do so. Therefore, the company that came second in the tendering was called in. Situations like this directly affect the cost, time and quality of the project and cause disruption.

The decision makers and the criteria that were defined and will be used in this application are shown below. The group comprises five DMs from different departments in the company: a technical engineer (D1); a quality engineer (D2); a security and environment engineer (D3); a budget manager (D4); and a contract manager (D5).

After that, the qualifiers and bid winners criteria to be used in the selection process are presented. The qualifier criteria are the following: (1) Responsibility, which is related to the bidder holding a quality certificate and having a security and environmental policy; (2) General experience - all companies should have at least five years' experience in construction projects. As bid winners' criteria, the following were defined:

- Cost: this is measured in an objective way by examining the average cost deviation for which the company was responsible on projects it has carried out in the last 5 years in relation to the planned cost;
- Culture: This is related to the willingness to incorporate new ideas and concepts;
- Design: This is in the sense of involving the construction companies and subcontractors who may have the skills needed to help the project designers in their activities;
- Quality: This aims to ensure that the quality of the project is effectively translated into a physical quality, furnishing a structure which has low operational and maintenance costs;
- Time: This corresponds to the total period of the project and construction phase, to the skill of planning correctly and finalizingactivities in accordance with the deadlines laid down by the client. It is measured in an objective way by the average of the deviations to schedules for which the company was responsible on projects it has carried out in the last 5 years in relation to the planned schedule;
- Experience: This considers the experience of key people who will work on the project, measured in terms of the engineers and project designers' years of experience.

The evaluations of culture, design and quality criteria are measured in a subjective way. The potential suppliers must answer a questionnaire to provide information for evaluating these criteria. Evaluation is made on a 5-point scale in which 1 is the worst evaluation and 5 is the best one.

After defining the criteria, their weights must be defined. As was presented earlier, each DM establishes his or her own weights for each criterion. With all the values, the variations occurring in the weight of each criterion are verified and a mean value for the weight is determined with a variation that encompasses the preferences of all DMs. Thereafter, these weights are normalized (Table 1).

Table 1. Table of normalized weights of criteria

	w_j	$\pm\Theta$
Cr1	0.118	30%
Cr2	0.176	30%
Cr3	0.294	50%
Cr4	0.176	10%
Cr5	0.118	10%
Cr6	0.118	10%

Since the criteria are defined, the briefing meeting can take place. All interested consultants must participate in this meeting, and bring the documentation regarding certifications and experience required as qualifier criteria. The whole selection process is explained at this meeting.

Next, the documentation delivered by the companies is evaluated. This is the filter stage. Six companies may go on and remain in the selection process.

At this time, each of these six companies receives a questionnaire that encloses aspects related to the qualitative criteria defined previously. The participants must return the questionnaire within the deadline of one week and provide information about the person(s) who answered the questions. Based on the information collected from the questionnaires, the companies are evaluated regarding their culture, quality and design - the evaluation is converted into a numerical scale as shown previously.

The decision matrix with all six companies evaluated by these criteria is given in Table 2.

Table 2. Decision matrix

	Cr_1	Cr_2	Cr_3	Cr_4	Cr_5	Cr_6
Company 1	0.10	4	5	4	0.10	8
Company 2	0.15	2	2	4	0.05	20
Company 3	0.15	1	2	3	-0.05	15
Company 4	0.03	1	1	4	0.15	6
Company 5	-0.03	4	4	3	0.20	10
Company 6	0.05	3	5	3	0.10	5

So as to apply the PROMETHEE II method, it is necessary to determine the preference functions for each criterion and the respective preference and/or indifference thresholds. Then, a joint discussion is carried out with the DMs and the analyst and the preference functions are established, in an interactive way. They all agree with the preferences functions and parameters that were chosen for each criterion. The only divergence between them was among the intercriteria information, since they are from different departments and give different importance for each aspect evaluated. But they not diverge in the way in which the alternatives are compared in each criterion. Then, for the criteria of culture, design and quality, type 1 preference functions were chosen. For the others, these functions are shown in 3.

Table 3. Preference functions

Criterion	Preference function	Parameter(s)

Cost	Type V	q = 0.02
		p=0.05

Time	Type III	p=0.05
Experience	Type III	p=2

With all this information, the PROMETHEE II method is applied and the net flows are obtained, as shown in Table 4.

Table 4. Net flows of the alternatives

Companies	Netflows
Company 1	0.44
Company 2	0.05
Company 3	-0.27
Company 4	-0.44
Company 5	0.12
Company 6	0.11

Since the net flows were obtained, the PROMETHEE VI is applied through the use of PROMCALC software. It can be seen from Fig. 2, that the alternative of best compromise is a_1, i.e., Company 1, since it is the only alternative that is in the direction of the decision axis for the whole set of weights contained in the zone stipulated.

PROMETHEE VI could also be considered as a tool for sensibility analysis, since it permits the region of parameter variation in the GAIA plan to be visualized and verifies whether the decision axe is still in the same direction of the alternative selected.

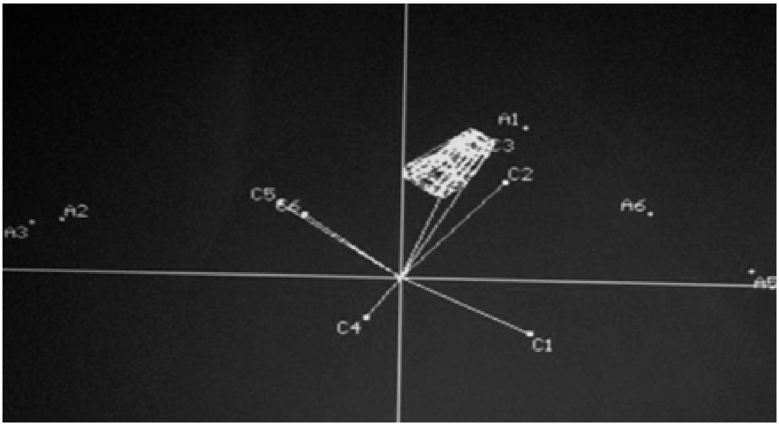


Fig. 2. PROMÉTHÉE VI - GAIA plane [30]

6 Concluding Remarks

The study conducted is related to selecting suppliers for a private sector company, involving multiple decision makers. It was found that the process used for selecting suppliers is a decisive event for the success of the construction project. Although there was a group of DMs, their preferences present little divergence.

A structure was presented for selecting suppliers that involved a group of DMs and one that deals in an appropriate way with their preferences. On adopting the structure, a considerable amount of time must be made available, since it is necessary to define properly the criteria involved and the criteria weights to evaluate them so as to proceed with the evaluation.

Although this study is contextualized for selecting a supplier for a construction project, the structure is suitable for selecting suppliers in all contexts where there is a group of DMs with the characteristics described previously.

Thus, it is expected that the supplier selected will be the one that is most committed to the objectives of the contractor company (identified through the criteria defined by the DMs) and will foster an integrated and cooperative relationship throughout the whole project.

Acknowledgements. This study was partially supported by CNPq (National Council for Scientific and Technological Development) to whom the authors are grateful.

References

1. Lysons, K., Farrington, B.: Purchasing and Supply Chain Management. Prentice Hall, Harlow (2006)
2. Gonzallez, M.E., Quesada, G., Monge, C.A.: Determining the importance of the supplier selection process in manufacturing: a case study. Int. J. of Physical Distrib. & Logistics Manag. 34, 492–504 (2004)

3. Dickson, G.: An analysis of vendor selection systems and decisions. J. Purch. 2, 5–17 (1966)
4. Dulmin, R., Mininno, V.: Supplier selection using a multi-criteria decision aid method. J. Purch. & Supply Manag. 9, 177–187 (2003)
5. Araz, C., Ozkarahan, I.: Supplier evaluation and management system for strategic sourcing based on a new multicriteria sorting procedure. Int. J. Prod. Econ. 106, 585–606 (2007)
6. De Boer, L., Wegen, L.V., Telgen, J.: Outranking methods in support of supplier selection. Eur. J. Purch. & Supply Manag. 4, 109–118 (1998)
7. Tama, M.C., Tummala, V.M.: An application of the AHP in vendor selection of a telecommunications system. Omega 29, 171–182 (2001)
8. Handfield, R., et al.: Applying environmental criteria to supplier assessment: A study in the application of the Analytical Hierarchy Process. Eur. J. Oper. Res. 141, 70–87 (2002)
9. Levary, R.R.: Ranking foreign suppliers based on supply risk. Supply Chain Manag.: an Int. J. 12, 392–394 (2007)
10. Almeida, A.T.: Multicriteria Decision Making on Maintenance: Spares and Contracts Planning. Eur. J. Oper. Res. 129, 235–241 (2001)
11. Saen, R.F.: A new mathematical approach for suppliers selection: accounting for non-homogeneity is important. Appl. Math. Comput. 185, 84–95 (2007)
12. Ho, W., Xu, X., Dey, P.K.: Multicriteria decision making approaches for supplier evaluation and selection: a literature review. Eur. J. Oper. Res. 202, 16–24 (2010)
13. Almeida, A.T.: Multicriteria Modelling of Repair Contract Based on Utility and ELECTRE I Method with Dependability and Service Quality Criteria. Annals of Oper. Res. 138, 113–126 (2005)
14. Ordoobadi, S.M.: Development of a supplier selection model using fuzzy logic. Supply Chain Manag.: Int. J. 14, 314–327 (2009)
15. Almeida, A.T.: Multicriteria decision model for outsourcing contracts selection based on utility function and ELECTRE method. Comp. and Oper. Res. 34, 3569–3574 (2007)
16. Kahraman, C., Cebeci, U., Ulukan, Z.: Multi-criteria supplier selection using Fuzzy AHP. Log. Inf. Mang. 16, 382–394 (2003)
17. Wu, D.: Supplier selection in a fuzzy group setting: A method using grey related analysis and Dempster-Shafer theory. Expert Systems with Applications 36, 8892–8899 (2009)
18. Zhang, D., et al.: An novel approach to supplier selection based on vague sets group decision. Expert Systems with Applications 36, 9557–9563 (2009)
19. Singh, D., Tiong, R.: A fuzzy decision framework for contractor selection. J. Const. Eng. and Manag. 131, 62–70 (2005)
20. Pongpeng, J., Liston, J.: Contractor ability criteria: a view from the Thai construction industry. Construction Management and Economics 21, 267–282 (2003)
21. Al-Reshaid, K., Kartam, N.: Design-build pre-qualification and tendering approach for public projects. Int. J. Proj. Manag. 23, 309–320 (2005)
22. Fu, C., Yang, S.-L.: The group consensus based evidential reasoning approach for multiple attributive group decision analysis. Eur. J. Oper. Res. 206, 601–608 (2010)
23. Parreiras, R., Ekel, P., Martini, J., Palhares, R.: A flexible consensus scheme for multicriteria group decision making under linguistic assessments. Inf. Sciences. 180, 1075–1089 (2010)
24. Leyva-Lpez, J., Fernndez-Gonzalez, E.: A new method for group decision support based on ELECTRE III methodology. Eur. J. of Operational Research. 26, 14–27 (2003)

25. Vincke, P.: Multicriteria decision-aid. Wiley, Bruxelles (1992)
26. Brans, J.P., Vincke, P.H.: A preference ranking organization method, the PROMETHEE method for MCDM. Manag. Science. 31, 647–656 (1985)
27. Brans, J.P., Mareschal, B.: The PROMETHEE VI procedure: how to differentiate hard from soft multicriteria problems. J. Dec. Syst. 4, 213–223 (1985)
28. Mareschal, B., Brans, J.: Geometrical representation for MCDM: the GAIA procedure. Eur. J. Oper. Res. 34, 69–77 (1988)
29. Brans, J., Mareschal, B.: PROMÉTHÉE - GAIA: une méthodologie d'aide á la décision en présence de critéres multiples. Éditions de L'Université de Bruxelles, Bruxelles (2002)
30. Alencar, L., Almeida, A.: A model for selecting project team members using multicriteria group decision making. Pesquisa Operacional 30, 221–236 (2010)

Author Index