618 LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS

Vincent Barichard
Matthias Ehrgott
Xavier Gandibleux
Vincent T'Kindt
(Editors)

Multiobjective Programming and Goal Programming

Theoretical Results and Practical Applications

Springer

Lecture Notes in Economics
and Mathematical Systems 618

Vincent Barichard · Matthias Ehrgott
Xavier Gandibleux · Vincent T'Kindt (Eds.)

Multiobjective Programming and Goal Programming

Theoretical Results and Practical Applications

Springer

Dr. Vincent Barichard
University of Angers
LERIA
2 Boulevard Lavoisier
49045 Angers Cedex 01
France
vincent.barichard@uni-angers.fr

Prof. Xavier Gandibleux
University of Nantes
LINA, Lab. d'Informatique de Nantes
Altantique
2 rue de la Houssinière
BP 92208
44322 Nantes
France
xavier.gandibleux@univ-nantes.fr

Prof. Matthias Ehrgott
University of Auckland
Dept. Engineering Science
Auckland 1020
New Zealand
m.ehrgott@auckland.ac.nz

Prof. Vincent T'Kindt
Université Francois-Rabelais
de Tours
Laboratoire d'Informatique
64 Avenue Jean Portalis
37200 Tours
France
tkindt@univ-tours.fr

ISBN 978-3-540-85645-0 e-ISBN 978-3-540-85646-7
DOI 10.1007/978-3-540-85646-7

Lecture Notes in Economics and Mathematical Systems ISSN 0075-8442

Library of Congress Control Number: 2008936142

Cover design: SPi Technologies, India

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

MOPGP is an international conference series devoted to multi-objective programming and goal programming (MOP/GP). This conference brings together researchers and practitioners from different disciplines of Computer Science, Operational Research, Optimisation Engineering, Mathematical Programming and Multi-criteria Decision Analysis. Theoretical results and algorithmic developments in the field of MOP and GP are covered, including practice and applications of MOP/GP in real-life situations.

The MOP/GP international conferences are organised in a biennial cycle. The previous editions were held in United Kingdom (1994), Spain (1996), Canada (1998), Poland (2000), Japan (2002), and Tunisia (2004). The Seventh meeting (MOPGP'06) was organised in the Loire Valley (Center-West of France) by X. Gandibleux, (University of Nantes, chairman) and V. T'Kindt (University of Tours, co-chairman). The conference was hosted during three days (June 12–14, 2006) by the old city hall of Tours which is located in the city centre of Tours.

The conference comprised four plenary sessions (M. Ehrgott; P. Perny; R. Caballero and F. Ruiz; S. Oussedik) and six semi-plenary sessions (N. Jussien and V. Barichard; D. Corne and J. Knowles; H. Hoogeveen; M. Wiecek; E. Bampis; F. Ben Abdelaziz) and 82 regular talks. The (semi-)plenary speakers were invited, while the regular talks were selected by the international scientific committee composed of 61 eminent researchers on basis of a 4-pages abstract.

Out of 115 regular talks submitted from 28 countries, 75% were finally accepted, covering 25 countries. A very low no-show rate of 2% was recorded. One hundred and twenty-five participants attended the meeting, including academics and practitioners from companies such as Renault, Electricité de France, Ilog, and Airbus. The biggest delegations came from France (22 plus the 10 members of the local organising committee), Spain (21), USA (10), Japan (7), Germany (6), Tunisia (6), UK (6).

Traditionally, a post-conference proceedings volume is edited for the MOP/GP conferences. For MOPGP'06, the decision has been to publish the volume by Springer in the Lecture Notes in Economics and Mathematical Systems series,

edited by V. Barichard, M. Ehrgott, X. Gandibleux and V. T'Kindt. The authors who presented a talk during the conference were invited to submit a 10-page paper presenting the full version of their work.

Forty-two regular papers plus two invited papers have been submitted. All of them have been refereed according to the standard reviewing process, by members of the MOPGP'06 international scientific committee and other expert referees: E. Bampis E., V. Barichard, S. Belmokhtar, F. Ben Abdelaziz, R. Caballero, S. Chu, C. Coello Coello, X. Delorme, P. Dépincé, C. Dhaenens, K. Doerner, M. Ehrgott, F. Fernandez Garcia, J. Figueira, J. Fodor, X. Gandibleux, J. Gonzalez-Pachon, S. Greco, T. Hanne, C. Henggeler Antunes, K. Hocine, H. Hoogeeven, H. Ishubuchi, J. Jahn, A. Jaszkiewicz, N. Katoh, I. Kojadinovic, F. Le Huede, A. Lotov, A. Marmol, K. Mieettinen, J. Molina, H. Nakayama, P. Perny, A. Przybylski, C. Romero, S. Sayin, R. Steuer, M. Tamiz, C. Tammer, T. Tanino, V. T'Kindt, T. Trzaskalik, D. Tuyttens, D. Vanderpooten, L. Vermeulen-Jourdan, M. Wiecek, E. Zitzler.

Finally, 26 papers have been accepted covering eight main topics of the conference. With the relatively high number of talks submitted for the conference, 75% of which have been accepted, followed by an acceptance rate of 59% for full papers, a fairly high quality of the proceedings is guaranteed.We are sure that the readers of those proceedings will enjoy the quality of papers published in this volume, which is structured in five parts:

1. Multiobjective Programming and Goal-Programming
2. Multiobjective Combinatorial Optimization
3. Multiobjective Metheuristics
4. Multiobjective Games and Uncertainty
5. Interactive Methods and Applications.

We wish to conclude by saying that we are very grateful to the authors who submitted their works, to the referees for their detailed reviews, and more generally, to all those contributing to the organization of the conference, peoples, institutions, and sponsors.

Angers, Auckland, Nantes, Tours
October 2008

Vincent Barichard
Matthias Ehrgott
Xavier Gandibleux
Vincent T'Kindt

Contents

Contributors

Kim Allan Andersen Department of Business Studies, University of Aarhus, Fuglesangs Alle 4, DK-8210 Aarhus V, Denmark, kia@asb.dk

Mar Arenas Universidad de Oviedo. Avenida del Cristo s/n, 33006-Oviedo, Spain, mariamar@uniovi.es

Fouad Ben Abdelaziz College of engineering, University of Sharjah, PO Box 26666, Sharjah, UAE, foued.benabdelaz@isg. rnu.tn; fabdelaziz@aus.edu

Patrick Beullens Management Mathematics Group, Department of Mathematics, University of Portsmouth, Portsmouth, PO1 3HF, United Kingdom, patrick.beullens@port. ac.uk

Lorenz T. Biegler Deptartment of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA, biegler@cmu.edu

Amelia Bilbao Universidad de Oviedo. Avenida del Cristo s/n, 33006-Oviedo, Spain, ameliab@uniovi.es

Vincent Y. Blouin Department of Mathematical Sciences, Clemson University, SC 29634, USA, wmalgor@clemson.edu

Marcin Czupryna Decision Support and Analysis Division, Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland, mczupr@sgh.waw.pl

Esther Dopazo Facultad de Informática, Technical University of Madrid, Campus de Montegancedo, CP28660, Boadilla del Monte (Madrid), Spain, edopazo@fi.upm.es

Roman Efremov Rey Juan Carlos University, c/ Tulipán s/n, Móstoles, Madrid, 28933, Spain, roman.efremov@urjc.es

Matthias Ehrgott Department of Engineering Science, The University of Auckland, Private Bag 92019, Auckland Mail Centre, Auckland 1142, New Zealand, m.ehrgott@auckland.ac.nz

Matthias Ehrgott Laboratoire d'Informatique de Nantes Atlantique, FRE CNRS 2729 – Université de Nantes. 2, rue de la Houssiniére, BP92208, F 44322 Nantes Cedex 03, – France, matthias.Ehrgott@univ-nantes.fr

Gabriele Eichfelder Institute of Applied Mathematics, University of Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany, gabriele.eichfelder@am.uni-erlangen.de

Alexander Engau Department of Management Sciences, University of Waterloo, Canada, aengau@alumni.clemson.edu

Petri Eskelinen Helsinki School of Economics, P.O. Box 1210, FI-00101 Helsinki, Finland, petri.eskelinen@hse.fi

Georges M. Fadel Department of Mathematical Sciences, Clemson University, SC 29634, USA, wmalgor@clemson.edu

Francisco R. Fernández Departamento de Estadstica e Investigacion Operativa, Universidad de Sevilla. Tarfia s/n. 41012- Sevilla, Spain, fernande@us.es

Yan Fu Ford Motor Company MD 2115, SRL 2122, Dearborn, MI 48124, USA, yfu4@ford.com

Blas Galván IUSIANI, University of Las Palmas de Gran Canaria Edif. Polivalente. Campus de Tafira, CP 35017 Las Palmas, Spain, bgalvan@step.es

Xavier Gandibleux Laboratoire d'Informatique de Nantes Atlantique, FRE CNRS 2729 – Université de Nantes. 2, rue de la Houssiniére, BP92208, F 44322 Nantes Cedex 03, France, xavier.gandibleux@univ-nantes.fr

David Greiner IUSIANI, University of Las Palmas de Gran Canaria Edif. Polivalente. Campus de Tafira, CP 35017 Las Palmas, Spain, dgreiner@iusiani.ulpgc.es

Jussi Hakanen, Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), FI-40014, Finland, jhaka@mit.jyu.fi

Thomas Hanne Department of Information Systems, University of Applied Sciences Northwestern Switzerland, Riggenbachstrasse 16 – 4600 Olten, Switzerland, thomas.hanne@fhnw.ch

Brian J. Hunt Department of Mathematical Sciences, Clemson University, SC 29634, USA, wmalgor@clemson.edu

Mariano Jiménez Universidad del País Vasco. Plaza Oñati 1, 20018-San Sebastián, Spain, mariano.jimenez@ehu.es

Dylan Jones Management Mathematics Group, Department of Mathematics, University of Portsmouth, Portsmouth, PO1 3HF, United Kingdom, Dylan.Jones@port.ac.uk

Julien Jorge Laboratoire d'Informatique de Nantes Atlantique, FRE CNRS 2729 – Université de Nantes. 2, rue de la Houssiniére, BP92208, F 44322 Nantes Cedex 03, France, julien.jorge@univ-nantes.fr

Ulrich Junker ILOG 1681, route des Dolines, F-06560 Valbonne, France, ujunker@ilog.fr

Bogumil Kaminski Decision Support and Analysis Division, Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland, bkamins@sgh.waw.pl

Yoshiaki Kawajiri Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA, kawajiri@cmu.edu

Saoussen Krichen LARODEC Laboratory, Institut Supérieur de Gestion, University of Tunis, 41 Rue de la Liberté, Le Bardo 2000, University of Tunis, Tunisia saoussen.krichen @isg.rnu.tn

Javier Larros Universitat Polit'ecnica de Catalunya Jordi Girona 1-3, Edificio Omega 08034 Barcelona, Spain, larrosa@lsi.upc.edu

Xiaodong Li Management Mathematics Group, Department of Mathematics, University of Portsmouth, Portsmouth, PO1 3HF, UK, xiaodong.li@port.ac.uk

Patrick Llerena Joint Research Unit ULP-CNRS, BETA 61 Avenue de la Fort Noire, Strasbourg, France, pllerena@cournot.u-strasbg.fr

Mariano Luque University of Málaga, Calle Ejido 6, E-29071 Málaga, Spain, mluque@uma.es

Máximo Méndez Computer Science Department, University of Las Palmas de Gran Canaria Edif. de Informática y Matemáticas. Campus de Tafira, CP 35017 Las Palmas, Spain, mmendez@dis.ulpgc.es

Kaisa Miettinen Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014, University of Jyväskylä, Finland, kaisa.miettinen@jyu.fi

Luisa Monroy Departamento de Economía Aplicada III. Universidad de Sevilla. Avda. Ramón y Cajal 1, 41018-Sevilla, Spain, lmonroy@us.es

Hiroyuki Morita Economics, Osaka Prefecture University, 1-1, Gakuen-cho, Sakai, Osaka, Japan, morita@eco.osakafu-u.ac.jp

Amir Nafi Joint Research Unit in Public Utilities Management, 1 Quai Koch BP F61039, 67070 Strasbourg, France, anafi@engees.u-strasbg.fr

Takanobu Nakahara Economics, Osaka Prefecture University, 1-1, Gakuen-cho, Sakai, Osaka, Japan, dq401005@edu.osakafu-u.ac.jp

Lus Paquete Department of Computer Engineering, CISUC Centre for Informatics and Systems of the University of Coimbra, University of Coimbra, 3030-290 Coimbra, Portugal, paquete@dei.uc.pt

Silvia Poles ESTECO, Area Science Park - Padriciano, 99 34012 Trieste, Italy, silvia.poles@esteco.com

Daniele Pretolani Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, Via Amendola 2, I-42100 Reggio, Emilia, Italy, daniele.pretolani@unimore.it

Anthony Przybylski Laboratoire d'Informatique de Nantes Atlantique, FRE CNRS 2729 – Université de Nantes. 2, rue de la Houssinière, BP92208, F 44322 Nantes Cedex 03, France, anthony.przybylski@univ-nantes.fr

Lars Relund Nielsen Department of Genetics and Biotechnology, Research Unit of Statistics and Decision Analysis, University of Aarhus, P.O. Box 50, DK-8830 Tjele, Denmark, lars@relund.dk

Enrico Rigoni ESTECO, Area Science Park - Padriciano, 99 34012 Trieste, Italy, enrico.rigoni@esteco.com

Emma Rollon Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edificio Omega – 08034 Barcelona, Spain, erollon@lsi.upc.edu

Francisco Ruiz University of Málaga, Calle Ejido 6, E-29071 Málaga, Spain, rua@uma.es

Mauricio Ruiz-Tagle Facultad de Ciencias de la Ingeniera, Universidad Austral de Chile, General Lagos 2086, Campus Miraflores, Valdivia, Chile, mruiztag@uach.cl

Daniel Salazar IUSIANI, University of Las Palmas de Gran Canaria Edif. Polivalente. Campus de Tafira, CP 35017 Las Palmas, Spain, danielsalazaraponte@gmail.com

Marc Sevaux LESTER, University of South Brittany Center de recherche, BP 92116, F56321 Lorient, France, marc.sevaux@univ-ubs.fr

Vijay Singh Department of Mathematical Sciences, Clemson University, SC 29634, USA, wmalgor@clemson.edu

Thomas Stützle Université Libre de Bruxelles, CoDE, IRIDIA, CP194/6, Avenue Franklin Roosevelt 50, 1050 Brussels, Belgium, stuetzle@ulb.ac.be

Tomasz Szapiro Division of Decision Analysis and Support, Institute of Econometrics, Warsawa School of Economics, Al. Niepodleglosci 162, 02-554 Warszzawa, tszapiro@sgh.waw.pl

Mehrdad Tamiz Management Mathematics Group, Department of Mathematics, University of Portsmouth, Portsmouth, PO1 3HF, United Kingdom, patrick.beullens@port. ac.uk

Christiane Tammer Martin-Luther-University Halle Halle-Wittenberg, Institute of Mathematics, 06099 Halle, Germany, christiane.tammer@mathematik.uni-halle.de

Tetsuzo Tanino Osaka University, Suita, Osaka 565-0871, Japan, tanino@eei.eng.osaka-u.ac.jp

Ma Victoria Rodríguez Universidad de Oviedo. Avenida del Cristo s/n, 33006-Oviedo, Spain, vrodri@uniovi.es

Caty Werey Joint Research Unit in Public Utilities Management, 1 Quai Koch BP F61039, 67070 Strasbourg, France, cwerey@engees.u-strasbg.fr

Margaret M. Wiecek Departments of Mathematical Sciences and Mechanical Engineering, Clemson University, Clemson, SC 29634, USA,wmalgor@clemson.edu

A Constraint Method in Nonlinear Multi-Objective Optimization

Gabriele Eichfelder

Abstract We present a new method for generating a concise and representative approximation of the (weakly) efficient set of a nonlinear multi-objective optimization problem. For the parameter dependent ε-constraint scalarization an algorithm is given which allows an adaptive controlling of the parameters–the upper bounds– based on sensitivity results such that equidistant approximation points are generated. The proposed method is applied to a variety of test-problems.

Keywords: Adaptive parameter control · Approximation · Multiobjective optimization · Scalarization · Sensitivity

1 Introduction

In many areas like economics, engineering, environmental issues or medicine the complex optimization problems cannot be described adequately by only one objective function. As a consequence multi-objective optimization which investigates optimization problems like

$$\begin{aligned} &\min\ f(x) = (f_1(x), \ldots, f_m(x))^\top \\ &\text{s.t. } x \in \Omega \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

with a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ with $m \in \mathbb{N}$, $m \geq 2$, and $\Omega \subseteq \mathbb{R}^n$ a closed set, is getting more and more important. For an introduction to multi-objective optimization see the books by Chankong and Haimes [3], Ehrgott [6], Hwang and Masud [14],

G. Eichfelder
Institute of Applied Mathematics, University of Erlangen-Nuremberg, Martensstr. 3,
91058 Erlangen, Germany
e-mail: Gabriele.Eichfelder@am.uni-erlangen.de

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

Jahn [16], Miettinen [21], Sawaragi et al. [24], and Steuer [27]. Further see the survey papers by Hillermaier and Jahn [13], and by Ruzika and Wiecek [23], with a focus on solution methods.

In general there is not only one best solution which minimizes all objective functions at the same time and the solution set, called efficient set, is very large. Especially in engineering tasks information about the whole efficient set is important. Besides having the whole solution set available the decision maker gets a useful insight in the problem structure. Consequently our aim is to generate a representative approximation of this set. The importance of this aim is also pointed out in many other works like, e. g. in [5, 9, 20, 26].

Thereby the information provided by the approximation set depends mainly on the quality of the approximation. With reference to quality criteria as discussed by Sayin in [25] we aim to generate almost equidistant approximation points. Further discussions on quality criteria for discrete approximations of the efficient set can be found, e. g. in [4, 18, 29, 31]. For reaching our target we use the well-known ε-constraint scalarization which is widely used in applications as it has easy to interpret parameters.

In this context the term of Edgeworth–Pareto (EP) optimal points as minimal solutions of (1) is very common which means that different points of the set $f(\Omega)$ are compared using the natural ordering introduced by the cone $\mathbb{R}^m_+$.

In Sect. 2 we give the basic notations in multi-objective optimization and we present the parameter dependent ε-constraint scalarization based on which we determine approximations of the (weakly) efficient set. The needed sensitivity results for an adaptive parameter control are given in Sect. 3. This results in the algorithm presented in Sect. 4 with a special focus on bi-objective optimization problems. In Sect. 5 we apply the algorithm on several test problems. Finally we conclude in Sect. 6 with an outlook on a generalization of the gained results.

2 Basic Notations and Scalarization

As mentioned in the introduction we are interested in finding minimal points of the multi-objective optimization problem (1) w. r. t. the natural ordering represented by the cone $\mathbb{R}^m_+$. A point $\bar{x} \in \Omega$ is called a $\mathbb{R}^m_+$-minimal point or an EP-minimal point of problem (1) if

$$(f(\bar{x}) - \mathbb{R}^m_+) \cap f(\Omega) = \{f(\bar{x})\}.$$

This is equivalent to that there exists no $x \in \Omega$ with $f_i(x) \leq f_i(\bar{x})$ for all $i = 1, \ldots, m$, and with $f_j(x) < f_j(\bar{x})$ for at least one $j \in \{1, \ldots, m\}$. The set of all EP-minimal points is denoted as $\mathscr{M}(f(\Omega))$. The set $\mathscr{E}(f(\Omega)) := \{f(x) \,|\, x \in \mathscr{M}(f(\Omega))\}$ is called efficient set. A point $\bar{x} \in \Omega$ is a weakly EP-minimal point if there is no point $x \in \Omega$ with $f_i(x) < f_i(\bar{x})$ for all $i = 1, \ldots, m$.

For obtaining single solutions of (1) we use the ε-constraint problem $(P_m(\varepsilon))$

$$\begin{aligned} &\min\ f_m(x) \\ &\text{s.t. } f_i(x) \le \varepsilon_i, \quad i = 1,\dots,m-1, \\ &\quad x \in \Omega \end{aligned} \tag{2}$$

with the upper bounds $\varepsilon = (\varepsilon_1,\dots,\varepsilon_{m-1})^\top$. This scalarization has the important properties that every EP-minimal point can be found as a solution of $(P_m(\varepsilon))$ by an appropriate parameter choice and every solution $\bar{x}$ of $(P_m(\varepsilon))$ is at least weakly EP-minimal.

For a discussion of this method see [3,6,10,21,27]. For the choice of the parameter ε Steuer [27] proposes a procedure based on a trial-and-error process. Sensitivity considerations w. r. t. the parameter ε are already done in [21] and [3]. There the Lagrange multipliers were interpreted as trade-off information. Based on this Chankong and Haimes [3] present the surrogate worth trade-off method. This procedure starts with the generation of a crude approximation of the efficient set by solving $(P_m(\varepsilon))$ for the parameters ε chosen from an equidistant grid. In an interactive process the decision maker chooses the preferred solution with the help of the trade-off information.

Solving problem $(P_m(\varepsilon))$ for various parameters ε leads to various (weakly) efficient points and hence to an approximation of the efficient set. Thereby we use the following definition of an approximation (see [11, p. 5]).

Definition 1. A finite set of points $A \subseteq f(\Omega)$ is called an approximation of the efficient set $\mathscr{E}(f(\Omega))$ of (1) if for all approximation points $y^1, y^2 \in A$, $y^1 \neq y^2$, it holds $y^1 \notin y^2 + \mathbb{R}^m_+$ and $y^2 \notin y^1 + \mathbb{R}^m_+$, i. e. the points in A are non-dominated w. r. t. the natural ordering.

Analogously we speak of an approximation of the weakly efficient set of (1) if for all $y^1, y^2 \in A$, $y^1 \neq y^2$ it holds $y^1 \notin y^2 + \operatorname{int}(\mathbb{R}^m_+)$ and $y^2 \notin y^1 + \operatorname{int}(\mathbb{R}^m_+)$.

Moreover we speak of an equidistant approximation with a distance of $\alpha > 0$ if for all $y \in \mathscr{E}(f(\Omega))$ there exists a point $\bar{y} \in A$ with

$$\|y - \bar{y}\| \le \frac{\alpha}{2}$$

(with $\|\cdot\|$ an arbitrary norm). Using the notation of Sayin [25] this corresponds to a coverage error of $\frac{\alpha}{2}$. Furthermore let

$$\min_{\substack{y^1, y^2 \in A \\ y^1 \neq y^2}} \|y^1 - y^2\| = \alpha,$$

i. e. let the uniformity level be α.

We summarize these conditions for the case of a bi-objective optimization problem, i. e. $m = 2$, by the following: Let $A = \{y^1,\dots,y^N\}$ be an approximation of the (weakly) efficient set of (1) with (weakly) efficient points. We speak of an equidistant approximation with the distance α if for consecutive (neighboured) approximation points, (e. g. ordered w. r. t. one coordinate in increasing order), it holds $\|y^{l+1} - y^l\| = \alpha$ for $l = 1\dots,N-1$.

3 Sensitivity Results

For controlling the choice of the parameter ε such that the generated points $f(x(\varepsilon))$ (with minimal solution $x(\varepsilon)$ of $(P_m(\varepsilon))$ w. r. t. the parameter ε) result in an equidistant approximation, we investigate the dependence of the minimal-value function of the problem $(P_m(\varepsilon))$ on the parameter. This is already done for a more general scalarization approach in [7]. Here, we apply these results on the ε-constraint method.

We suppose the constraint set Ω is given by

$$\Omega = \{x \in \mathbb{R}^n \mid g_j(x) \geq 0,\ j = 1,\ldots,p,\ h_k(x) = 0,\ k = 1,\ldots,q\}$$

with continuous functions $g_j \colon \mathbb{R}^n \to \mathbb{R}$, $j = 1,\ldots,p$, $h_k \colon \mathbb{R}^n \to \mathbb{R}$, $k = 1,\ldots,q$. We denote the index sets of active non-degenerate, active degenerate, and inactive constraints g_j as J^+, J^0, J^- respectively. Equally we set I^+, I^0 and I^- regarding the constraints $\varepsilon_i - f_i(x) \geq 0$ ($i \in \{1,\ldots,m-1\}$). The following result is a conclusion from a theorem by Alt in [1] as well as an application of a sensitivity theorem by Luenberger [19].

Theorem 1. *Suppose x^0 is a local minimal solution of the so-called reference problem $(P_m(\varepsilon^0))$ with Lagrange multipliers $(\mu^0, \nu^0, \xi^0) \in \mathbb{R}^{m-1}_+ \times \mathbb{R}^p_+ \times \mathbb{R}^q$ and there exists $\gamma > 0$ with f, g, h twice continuously differentiable on an open neighbourhood of the closed ball $B_\gamma(x^0)$. Let the gradients of the active constraints be linearly independent and let the second order sufficient condition for a local minimum of $(P_m(\varepsilon^0))$ hold in x^0, i. e. there exists some $\beta > 0$ with*

$$x^\top \nabla_x^2 \mathscr{L}(x^0, \mu^0, \nu^0, \xi^0, \varepsilon^0) x \geq \beta \|x\|^2 \text{ for all}$$

$$x \in \{x \in \mathbb{R}^n \mid \nabla_x f_i(x^0)^\top x = 0,\ \forall i \in I^+,\ \nabla_x g_j(x^0)^\top x = 0,\ \forall j \in J^+,$$

$$\nabla_x h_k(x^0)^\top x = 0,\ \forall k = 1,\ldots,q\}$$

for the Hessian of the Lagrange-function at x^0.

Then x^0 is a local unique minimal solution of $(P_m(\varepsilon^0))$, the associated Lagrange multipliers are unique, and there exists a $\delta > 0$ and a neighbourhood $N(\varepsilon^0)$ of ε^0 such that the local minimal-value function $\tau^\delta \colon \mathbb{R}^{m-1} \to \overline{\mathbb{R}}$,

$$\tau^\delta(\varepsilon) := \inf\{f_m(x) \mid f_i(x) \leq \varepsilon_i,\ i = 1,\ldots,m-1,\ g_j(x) \geq 0,\ j = 1,\ldots,p,$$
$$h_k(x) = 0, k = 1,\ldots,q,\ x \in B_\delta(x^0)\}$$

is differentiable on $N(\varepsilon^0)$ with

$$\frac{\partial \tau^\delta(\varepsilon^0)}{\partial \varepsilon_i} = -\mu_i^0 \text{ for } i = 1,\ldots,m-1.$$

Chankong and Haimes present a similar result [3, p. 160] assuming non-degeneracy. Regarding the assumptions of the theorem the active non-degenerate constraints remain active and the inactive constraints equally in a neighbourhood of ε^0.

We use the results of Theorem 1 for controlling the choice of the parameter ε. Therefore the Lagrange multipliers μ_i^0 to the parameter dependent constraints are needed. Thus a numerical method for solving $(P_m(\varepsilon^0))$ is necessary which provides these Lagrange multipliers. If the problem $(P_m(\varepsilon^0))$ is non-convex a numerical procedure for finding global optimal solutions has to be applied. If this procedure does not provide the Lagrange multipliers the global solution (or an approximation of it) can be used as a starting point for an appropriate local solver.

The assumptions of Theorem 1 are not too restrictive. In many applications it turns out that the efficient set is smooth, see e. g. [2,8]. The efficient set corresponds directly to the solutions and minimal values of the ε-constraint scalarization for varying parameters. Thus differentiability of the minimal-value function w. r. t. the parameters can be presumed in many cases.

4 Controlling of Parameters and Algorithm

Here we concentrate on the bi-objective case $m = 2$. A generalization to the case $m \geq 3$ for generating local equidistant points can be done easily but for an equidistant approximation of the whole efficient set problems occur as discussed in [7]. For example for $m = 2$ we can restrain the parameter set by solving the scalar optimization problems $\min_{x\in\Omega} f_1(x) =: f_1(\bar{x}^1)$ and $\min_{x\in\Omega} f_2(x) =: f_2(\bar{x}^2)$. Then, if $\bar{x}$ is EP-minimal for (1) there exists a parameter ε so that $\bar{x}$ is a minimal solution of $(P_2(\varepsilon))$ with $f_1(\bar{x}^1) \leq \varepsilon \leq f_1(\bar{x}^2)$ (see [7, 8, 15]). This cannot be transferred to the case with three or more objective functions. We comment on the mentioned generalization to three and more objectives for generating local equidistant approximation points at the end of this Section.

We assume we have solved $(P_2(\varepsilon^0))$ for a special parameter ε^0 with an at least weakly EP-minimal solution $x(\varepsilon^0) = x^0$ and that the assumptions of Theorem 1 are satisfied. The problem $(P_2(\varepsilon^0))$ is called the reference problem. The point $f(x^0)$ is an approximation point of the (weakly) efficient set. Now we are looking for a parameter ε^1 with

$$\|f(x(\varepsilon^1)) - f(x^0)\| = \alpha \tag{3}$$

(e. g. $\|\cdot\| = \|\cdot\|_2$ or any other norm) for a given value $\alpha > 0$. Further we assume that the constraint $f_1(x) \leq \varepsilon^0$ is active (if not, a parameter $\tilde{\varepsilon}^0$ with $f_1(x^0) = \tilde{\varepsilon}^0$ and x^0 minimal solution of $(P_2(\tilde{\varepsilon}^0))$ can be determined easily). Since, under the assumptions of Theorem 1, active constraints remain active, we can presume $f_1(x(\varepsilon^1)) = \varepsilon^1$ for the minimal solution $x(\varepsilon^1)$ of $(P_2(\varepsilon^1))$.

We use the derivative of the minimal-value function for a Taylor approximation. We assume it is possible to presume smoothness of the minimal-value function, see the comment at the end of Sect. 3. Then we get the following local approximation

$$f_2(x(\varepsilon^1)) \approx f_2(x^0) - \mu^0(\varepsilon^1 - \varepsilon^0). \tag{4}$$

As a consequence the relation (3) is approximately satisfied by setting

$$\varepsilon^1 = \varepsilon^0 \pm \alpha \cdot \left(\sqrt{1 + (\mu^0)^2} \right)^{-1}.$$

This results in the following algorithm.

Algorithm for the case $m = 2$:

Step 1: Choose a desired distance $\alpha > 0$ between approximation points. Choose M so that $M > f_1(x)$ for all $x \in \Omega$. Solve $(P_2(M))$ with minimal solution x^1 and Lagrange multipliers (μ^1, ν^1, ξ^1). Set $\varepsilon^1 := f_1(x^1)$ and $l := 1$.

Step 2: Solve $\min\limits_{x \in \Omega} f_1(x)$ with minimal solution x^E.

Step 3: Set

$$\varepsilon^{l+1} := \varepsilon^l - \frac{\alpha}{\sqrt{1 + (\mu^l)^2}} \text{ and } l := l + 1.$$

Step 4: If $\varepsilon^l \geq f_1(x^E)$ solve $(P_2(\varepsilon^l))$ with minimal solution x^l and Lagrange multipliers (μ^l, ν^l, ξ^l) and go to step 3. Otherwise stop.

This algorithm leads to an approximation $\{x^1, \ldots, x^{l-1}, x^E\}$ of the set of weakly EP-minimal points and so in an approximation $\{f(x^1), \ldots, f(x^{l-1}),\ f(x^E)\}$ of the weakly efficient set with points with a distance of approximately α. Solving problem $(P_2(M))$ in Step 1 is equivalent to solve $\min_{x \in \Omega} f_2(x)$.

Often the distance α which is needed in Step 3 results from the considered application. Otherwise, the following guideline can be used: for a desired number of approximation points $N \in \mathbb{N}$ choose a value $\alpha \leq \|f(x^E) - f(x^1)\| \cdot (N-1)^{-1}$.

The described parameter control can be generalized to $m \geq 2$ using the described procedure for determining local equidistant approximation, e. g. for doing a refinement of an approximation around a single approximation point $f(x^0)$. Then we set $\varepsilon^0 := (f_1(x^0), \ldots, f_{m-1}(x^0))$. The point x^0 is a minimal solution of $(P_m(\varepsilon^0))$. Let $v \in \mathbb{R}^{m-1}$ be a direction so that we are looking for a new parameter $\varepsilon^1 := \varepsilon^0 + s \cdot v$ for a $s \in \mathbb{R}$ such that (3) is satisfied. We can presume $f_i(x(\varepsilon^1)) = \varepsilon_i^1$ for $i = 1, \ldots, m-1$, as active constraints remain active, and further $f_m(x(\varepsilon^1)) \approx f_m(x^0) - s\,\mu^{0\top} v$ as in (4). This results in

$$|s| = \alpha \cdot \left\| \begin{pmatrix} v \\ -\mu^{0\top} v \end{pmatrix} \right\|^{-1}.$$

For the direction v we can choose e. g. the $m-1$ unit vectors in $\mathbb{R}^{m-1}$. For finding even more refinement points additional parameters can be determined for $\varepsilon^1 = \varepsilon^0 \pm 2 \cdot s \cdot v$ and so on (see [8]).

5 Numerical Results

In this section we apply the proposed adaptive parameter control to some test problems. Here we always choose the Euclidean norm. Applying the algorithm to the following easy example by Hazen [12, p.186]

$$\begin{aligned} &\min \left(\tfrac{x_1^2}{2} + x_2^2 - 10x_1 - 100 \, , \, x_1^2 + \tfrac{x_2^2}{2} - 10x_2 - 100 \right)^\top \\ &\text{s.t. } x \in \mathbb{R}^2 \end{aligned}$$

leads to the almost equidistant approximation shown in Fig. 1, (a). The chosen parameters are drawn as points $(\varepsilon, 10) \in \mathbb{R}^2$, too. For comparison the unsatisfactory result of the common ε-constraint method with an equidistant choice of parameters can be seen in Fig. 1, (b).

The algorithm works on multi-objective optimization problems with a non-convex image set, too, as it is demonstrated on the following problem by Tanaka [28] (Fig. 1, (c)):

$$\begin{aligned} &\min \, (x_1 \, , \, x_2)^\top \\ &\text{s.t. } x_1^2 + x_2^2 - 1 - 0.1 \cos\left(16 \arctan \tfrac{x_1}{x_2}\right) \geq 0, \\ &\quad (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \\ &\quad x_1, x_2 \in [0, \pi]. \end{aligned}$$

Here, the efficient set is even non-connected. For this problem we minimized the objective f_1 and solved the scalar optimization problems $(P_1(\varepsilon))$.

For comparison with the well-known wide-spread scalarization approach of the weighted-sum method [30], $\min_{x \in \Omega} w_1 f_1(x) + w_2 f_2(x)$, with weights $w_1, \; w_2 \in [0,1]$, $w_1 + w_2 = 1$, we consider the problem

$$\begin{aligned} &\min \left(\sqrt{1 + x_1^2} \, , \, x_1^2 - 4x_1 + x_2 + 5 \right)^\top \\ &\text{s.t. } x_1^2 - 4x_1 + x_2 + 5 \leq 3.5, \\ &\quad x_1 \geq 0, \; x_2 \geq 0. \end{aligned}$$

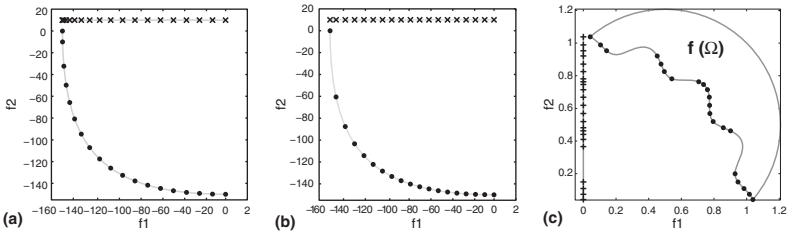


Fig. 1 Approximation of the efficient set of the test problem by Hazen (**a**) with adaptively controlled and (**b**) with equidistant parameters. (**c**) Approximation for the test problem by Tanaka

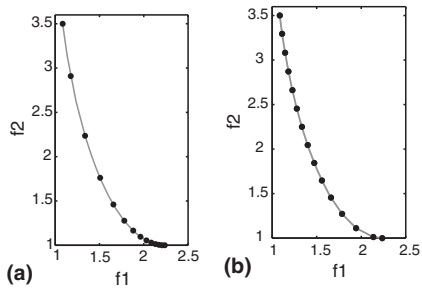


Fig. 2 (**a**) Approximation with the weighted-sum and (**b**) with the new method

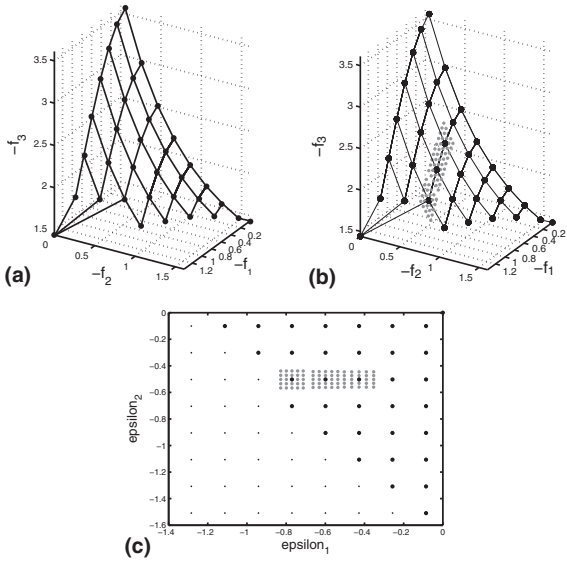


Fig. 3 (**a**) Approximation with the ε-constraint-method. (**b**) Refined approximation. (**c**) Chosen parameters

By an equidistant variation of the weights we get the 15 approximation points shown in Fig. 2, (a) with a highly non-uniform distribution. Using instead the method proposed here we get the representative approximation with 15 points of Fig. 2, (b).

Finally we consider a multi-objective test problem with three objectives which is a modified version of a problem by Kim and de Weck [17] with a non-convex image set:

$$\begin{aligned} &\min \left(-x_1\,,\,-x_2\,,\,-x_3^2\right)^\top \\ &\text{s.t. } -\cos(x_1)-\exp(-x_2)+x_3 \leq 0, \\ &\qquad 0 \leq x_1 \leq \pi,\ x_2 \geq 0,\ x_3 \geq 1.2. \end{aligned}$$

Applying the ε-constraint method for determining a crude approximation by using parameters on an equidistant grid leads to Fig. 3, (a).

The decision maker can now choose some especially interesting points. Then, in a second step, a refinement around these points now using sensitivity information can be done for gaining locally almost equidistant points in the image space (see Fig. 3, (b)) as described at the end of Sect. 4. The chosen parameters are drawn in Fig. 3, (c). It can well be seen that the distance between the refinement parameters varies depending on the sensitivity information.

In [7, 8] more test problems are solved. Furthermore a relevant bi- and tri-objective application problem from intensity modulated radiotherapy planning with more than 17,000 constraints and 400 variables is computed with the new method introduced here.

6 Conclusion

We have presented a new method for controlling the parameters of the ε-constraint method adaptively for generating (locally) concise but representative approximations of the efficient set of non-linear multi-objective optimization problems. Only by using sensitivity information which we get with no additional effort by solving the scalar optimization problems and making use of the byproduct of the Lagrange multipliers we can control the distances of the approximation points of the efficient set.

A generalization using the scalarization according to Pascoletti and Serafini [22] which allows to deal with arbitrary partial orderings in the image space introduced by a closed pointed convex cone is done in [7].

The author is grateful to the referees for their valuable comments and suggestions.

References

1. Alt W (1991) Parametric optimization with applications to optimal control and sequential quadratic programming. Bayreuther Mathematische Schriften 35:1–37
2. Bijick E, Diehl D, Renz W (2006) Bikriterielle Optimierung des Hochfrequenzfeldes bei der Magnetresonanzbildung. In: Küfer KH et al (eds) Multicriteria decision making and fuzzy systems, theory, methods and applications. Shaker, Aachen, pp 85–98
3. Chankong V, Haimes YY (1983) Multiobjective decision making. Elsevier, Amsterdam
4. Collette Y, Siarry P (2005) Three new metrics to measure the convergence of metaheuristics towards the Pareto frontier and the aesthetic of a set of solutions in biobjective optimization. Comput Oper Res 32(4):773–792
5. Das I, Dennis J (1998) Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657
6. Ehrgott M (2000) Multicriteria optimisation. Springer, Berlin
7. Eichfelder G (2006) Parametergesteuerte Lösung nichtlinearer multikriterieller Optimierungsprobleme (in German). Dissertation, University of Erlangen-Nuremberg, Germany

8. Eichfelder G (2006) ε-constraint method with adaptive parameter control and an application to intensity-modulated radiotherapy. In: Küfer KH et al (eds) Multicriteria decision making and fuzzy systems, theory, methods and applications. Shaker, Aachen, pp 25–42
9. Fliege J (2006) An efficient interior-point method for convex multicriteria optimization problems. Math Oper Res 31(4):825–845
10. Haimes YY, Lasdon LS, Wismer DA (1971) On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans Syst Man Cybern 1:296–297
11. Hansen MP, Jaszkiewicz A (1998) Evaluating the quality of approximations to the non-dominated set. IMM Technical University of Denmark, Denmark, IMM-REP-1998-7
12. Hazen GB (1988) Differential characterizations of nonconical dominance in multiple objective decision making. Math Oper Res 13(1):174–189
13. Hillermeier C, Jahn J (2005) Multiobjective optimization: survey of methods and industrial applications. Surv Math Ind 11:1–42
14. Hwang C-L, Masud AS (1979) Multiple objective decision making - methods and applications. A state-of-the-art survey. Springer, Berlin
15. Jahn J, Merkel A (1992) Reference point approximation method for the solution of bicriterial nonlinear optimization problems. J Optim Theory Appl 74(1): 87–103
16. Jahn J (2004) Vector optimization: theory, applications and extensions. Springer, Berlin
17. Kim IY, de Weck O (2005) Adaptive weighted sum method for bi-objective optimization. Struct Multidisciplinary Optim 29:149–158
18. Knowles J, Corne D (2002) On metrics for comparing non-dominated sets. Proceedings of the World Congress on Computational Intelligence, pp 711–716
19. Luenberger DG (1973) Introduction to linear and nonlinear programming. Addison-Wesley, Reading, MA
20. Messac A, Mattson CA (2004) Normal constraint method with guarantee of even representation of complete Pareto frontier. AIAA J 42(1):2101–2111
21. Miettinen KM (1999) Nonlinear multiobjective optimization. Kluwer, Boston
22. Pascoletti A, Serafini P (1984) Scalarizing vector optimization problems. J Optim Theory Appl 42(4):499–524
23. Ruzika S, Wiecek MM (2005) Approximation methods in multiobjective programming. J Optim Theory Appl 126(3):473–501
24. Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, London
25. Sayin S (2003) Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Math Program 87(A):543–560
26. Schandl B, Klamroth K, Wiecek MM (2001) Norm-based approximation in bicriteria programming. Comput Optim Appl 20(1):23–42
27. Steuer RE (1986) Multiple criteria optimization: theory, computation, and application. Wiley, New York
28. Tanaka M (1995) GA-based decision support system for multi-criteria optimization. Proc Int Conf Syst Man Cybern 2:1556–1561
29. Van Veldhuizen DA (1999) Multiobjective evolutionary algorithms: classifications, analyses, and new innovation. Dissertation, Graduate School of Engineering, Air Force Institute of Technology Dayton, Ohio, USA
30. Zadeh L (1963) Optimality and non-scalared-valued performance criteria. IEEE Trans. Autom Control 8:59–60
31. Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and application. Dissertation, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland

The Attainment of the Solution of the Dual Program in Vertices for Vectorial Linear Programs

Frank Heyde, Andreas Löhne, and Christiane Tammer

Abstract This article is a continuation of Löhne A, Tammer C (2007: A new approach to duality in vector optimization. Optimization 56(1–2):221–239) [14]. We developed in [14] a duality theory for convex vector optimization problems, which is different from other approaches in the literature. The main idea is to embed the image space $\mathbb{R}^q$ of the objective function into an appropriate complete lattice, which is a subset of the power set of $\mathbb{R}^q$. This leads to a duality theory which is very analogous to that of scalar convex problems. We applied these results to linear problems and showed duality assertions. However, in [14] we could not answer the question, whether the supremum of the dual linear program is attained in vertices of the dual feasible set. We show in this paper that this is, in general, not true but, it is true under additional assumptions.

Keywords: Dual program · Linear programs · Multi-objective optimization · Vertices

1 Introduction

Vectorial linear programs play an important role in economics and finance and there have been many efforts to solve those problems with the aid of appropriate algorithms. There are several papers on variants of the simplex algorithm for the multiobjective case, see e.g. Armand and Malivert [1], Ecker et al. [2], Ecker and Kouada [3], Evans and Steuer [5], Gal [6], Hartley [8], Isermann [9], Philip [16, 17], Yu and Zeleny [21], and Zeleny [22]. However, neither of these papers consider a dual simplex algorithm, which is in scalar linear programming a very important tool from the theoretical as well as the practical point of view. In the paper by Ehrgott,

C. Tammer (✉)
Martin-Luther-University Halle-Wittenberg, Institute of Mathematics, 06099 Halle, Germany
e-mail: christiane.tammer@mathematik.uni-halle.de

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

Puerto and Rodríguez-Chía [4] it was mentioned that "multi-objective duality theory cannot easily be used to develop a [dual or primal-dual simplex] algorithm". It is therefore our aim to consider an alternative approach to duality theory, which is appropriate for a dual simplex algoritm. This approach differs essentially from those in the literature (cf. Yu and Zeleny [21], Isermann [9], and Armand and Malivert [1]). In [14] we developed the basics of our theory and showed weak and strong duality assertions. The main idea is to embed the image space $\mathbb{R}^q$ of the objective function into a complete lattice, in fact into the space of self-infimal subsets of $\mathbb{R}^q \cup \{-\infty, +\infty\}$. As a result, many statements well-known from the case of scalar linear programming can be expressed analogously. However, in order to develop a dual simplex algorithm, it is important to have the property that the supremum of the dual problem is attained in vertices of the dual feasible set. This ensures that we only have to search a finite subset of feasible points. However, in [14] we could not show this attainment property, which is therefore the main subject of the present paper.

After a short introduction into the notation and the results of [14] we show that the attainment property is not true, in general. But, supposing some relatively mild assumptions, we can prove that for one of the three types of problems considered in [14], the supremum is indeed attained in vertices. This result is obtained by showing a kind of quasi-convexity of the (set-valued) dual objective function, which together with its concavity is a replacement for linearity. We further see that it is typical that the supremum is not attained in a single vertex (like in the scalar case) but in a set of possibly more than one vertex.

The application of these result in order to develop a dual simplex algorithm is presented in a forthcoming paper.

2 Preliminaries

We start to introduce the space of self-infimal sets, which plays an important role in the following. For a more detailed discussion of this space see [14].

Let $C \subsetneq \mathbb{R}^q$ be a closed convex cone with nonempty interior. The set of *minimal* or *weakly efficient* points of a subset $B \subseteq \mathbb{R}^q$ (with respect to C) is defined by

$$\operatorname{Min} B := \{y \in B | \ (\{y\} - \operatorname{int} C) \cap B = \emptyset\}.$$

The *upper closure* (with respect to C) of $B \subseteq \mathbb{R}^q$ is defined to be the set

$$\operatorname{Cl}_+ B := \{y \in \mathbb{R}^q | \ \{y\} + \operatorname{int} C \subseteq B + \operatorname{int} C\}.$$

Before we recall the definition of infimal sets, we want to extend the upper closure for subsets of the space $\overline{\mathbb{R}}^q := \mathbb{R}^q \cup \{-\infty, +\infty\}$. For a subset $B \subseteq \overline{\mathbb{R}}^q$ we set

$$\operatorname{Cl}_+ B := \begin{cases} \mathbb{R}^q & \text{if} \quad -\infty \in B \\ \emptyset & \text{if} \quad B = \{+\infty\} \\ \{y \in \mathbb{R}^q | \ \{y\} + \operatorname{int} C \subseteq B + \operatorname{int} C\} & \text{else.} \end{cases}$$

Note that the upper closure of a subset of $\overline{\mathbb{R}}^q$ is always a subset in $\mathbb{R}^q$. The *infimal set* of $B \subseteq \overline{\mathbb{R}}^q$ (with respect to C) is defined by

$$\operatorname{Inf} B := \begin{cases} \operatorname{MinCl}_+ B & \text{if } \emptyset \neq \operatorname{Cl}_+ B \neq \mathbb{R}^q \\ \{-\infty\} & \text{if } \operatorname{Cl}_+ B = \mathbb{R}^q \\ \{+\infty\} & \text{if } \operatorname{Cl}_+ B = \emptyset. \end{cases}$$

This means that the *infimal set of B with respect to C* coincides essentially with the set of weakly efficient elements of the set $\operatorname{cl}(B+C)$ with respect to C. The supremal set of a set $B \subseteq \overline{\mathbb{R}}^q$ is defined analogously and is denoted by $\operatorname{Sup} B$. To this end, we have $\operatorname{Sup} B = -\operatorname{Inf}(-B)$. In the sequel we need the following assertions due to Nieuwenhuis [15]. For $B \subseteq \mathbb{R}^q$ with $\emptyset \neq B + \operatorname{int} C \neq \mathbb{R}^q$ it holds

$$\operatorname{Inf} B = \{y \in \mathbb{R}^q \mid y \notin B + \operatorname{int} C,\ \{y\} + \operatorname{int} C \subseteq B + \operatorname{int} C\}, \tag{1}$$

$$\operatorname{Inf} B \cap B = \operatorname{Min} B. \tag{2}$$

Let $\mathscr{I}$ be the family of all self-infimal subsets of $\overline{\mathbb{R}}^q$, i.e., all sets $B \subseteq \overline{\mathbb{R}}^q$ satisfying $\operatorname{Inf} B = B$. In $\mathscr{I}$ we introduce an order relation $\preccurlyeq$ as follows:

$$B_1 \preccurlyeq B_2 :\Longleftrightarrow \operatorname{Cl}_+ B_1 \supseteq \operatorname{Cl}_+ B_2.$$

As shown in [14], there is an isotone bijection j between the space $(\mathscr{I}, \preccurlyeq)$ and the space $(\mathscr{F}, \supseteq)$ of upper closed subsets of $\mathbb{R}^q$ ordered by set inclusion. Indeed, one can choose

$$j : \mathscr{I} \to \mathscr{F}, \quad j(\cdot) = \operatorname{Cl}_+(\cdot), \quad j^{-1}(\cdot) = \operatorname{Inf}(\cdot).$$

Note that j is also isomorphic for an appropriate definition of an addition and a multiplication by nonnegative real numbers. Moreover, $(\mathscr{I}, \preccurlyeq)$ is a complete lattice and for nonempty sets $\mathscr{B} \subseteq \mathscr{I}$ it holds [14, Theorem 3.5]

$$\inf \mathscr{B} = \operatorname{Inf} \bigcup_{B \in \mathscr{B}} B, \qquad \sup \mathscr{B} = \operatorname{Sup} \bigcup_{B \in \mathscr{B}} B.$$

This shows that the infimum and the supremum in $\mathscr{I}$ are closely related to the usual solution concepts in vector optimization.

In [14] we considered the following three linear vector optimization problems. As usual in vector optimization we use the abbreviation $f[S] := \bigcup_{x \in S} f(x)$.

$$(\mathrm{LP}^1) \qquad \bar{P} = \operatorname{Inf} M[S], \qquad S := \{x \in \mathbb{R}^n \mid Ax \geq b\},$$

$$(\mathrm{LP}^2) \qquad \bar{P} = \operatorname{Inf} M[S], \qquad S := \{x \in \mathbb{R}^n \mid x \geq 0,\ Ax \geq b\},$$

$$(\mathrm{LP}^3) \qquad \bar{P} = \operatorname{Inf} M[S], \qquad S := \{x \in \mathbb{R}^n \mid x \geq 0,\ Ax = b\},$$

where $M \in \mathbb{R}^{q \times n}, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$.

In the following let a vector $c \in \operatorname{int} C$ be fixed. In [14] we calculated the dual problems to (LP^i) $(i = 1,2,3)$ (depending on c) as

$$(\mathrm{LD}_c^1) \qquad \begin{cases} \bar{D}_c = \mathrm{Sup}\bigcup_{u\in T_c}\left(cu^Tb + \mathrm{Inf}(M-cu^TA)[\mathbb{R}^n]\right) \\ T_c := \left\{u\in\mathbb{R}^m |\ u\geq 0,\ \exists c^*\in B_c : A^Tu = M^Tc^*\right\}, \end{cases}$$

$$(\mathrm{LD}_c^2) \qquad \begin{cases} \bar{D}_c = \mathrm{Sup}\bigcup_{u\in T_c}\left(cu^Tb + \mathrm{Inf}(M-cu^TA)[\mathbb{R}^n_+]\right) \\ T_c := \left\{u\in\mathbb{R}^m |\ u\geq 0,\ \exists c^*\in B_c : A^Tu \leq M^Tc^*\right\}. \end{cases}$$

$$(\mathrm{LD}_c^3) \qquad \begin{cases} \bar{D}_c = \mathrm{Sup}\bigcup_{u\in T_c}\left(cu^Tb + \mathrm{Inf}(M-cu^TA)[\mathbb{R}^n_+]\right) \\ T_c := \left\{u\in\mathbb{R}^m |\ \exists c^*\in B_c : A^Tu \leq M^Tc^*\right\}. \end{cases}$$

where the compact and convex set $B_c := \{c^*\in -C^\circ |\ \langle c, c^*\rangle = 1\}$ is used to express the dual constraints. We observed in [14] that the set T_c in (LD_c^1)–(LD_c^3) is always a closed convex subset of $\mathbb{R}^m$ and, if C is polyhedral, then T_c is polyhedral, too. Moreover, we have shown the following duality result.

Theorem 1. [14] *It holds weak and strong duality between* $(\mathrm{LP}^{\mathrm{i}})$ *and* $(\mathrm{LD}_c^{\mathrm{i}})$ $(i = 1,2,3)$. *More precisely we have*

1. $\bar{D}_c = \bar{P} \subseteq \mathbb{R}^q$ *if* $S\neq\emptyset$ *and* $T_c\neq\emptyset$, *where "*Sup*" can be replaced by "*Max*" in this case,*
2. $\bar{D}_c = \bar{P} = \{-\infty\}$ *if* $S\neq\emptyset$ *and* $T_c = \emptyset$,
3. $\bar{D}_c = \bar{P} = \{+\infty\}$ *if* $S = \emptyset$ *and* $T_c\neq\emptyset$.

The following example [14] illustrates the dual problem and the strong duality. Moreover, in this example we have the attainment of the supremum of the dual problem in the (three) vertices of the dual feasible set T.

Example 1. [14] (see Fig. 1) Let $q = m = n = 2$, $C = \mathbb{R}^2_+$ and consider the problem (LP^2) with the data

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

The dual feasible set for the choice $c = (1,1)^T \in \mathrm{int}\mathbb{R}^2_+$ is $T_c = \{\, u_1, u_2 \geq 0 |\ u_1 + u_2 \leq 1/3\,\}$. The vertices of T_c are the points $v_1 = (0,0)^T$, $v_2 = (1/3,0)^T$ and $v_3 = (0,1/3)^T$. We obtain the values of the dual objective function at v_1, v_2, v_3 as $D_c(v_1) = \mathrm{bd}\mathbb{R}^2_+$, $D_c(v_2) = \left\{y\in\mathbb{R}^2 |\ y_1 + 2y_2 = 2\right\}$ and $D_c(v_3) = \{y\in\mathbb{R}^2 |\ 2y_1 + y_2 = 2\}$. We see that the three dual feasible points $v_1, v_2, v_3 \in T_c$ are already sufficient for strong duality.

3 Dual Attainment in Vertices

We start with an example that shows that the supremum of the dual problem is, in general, not attained in vertices of the dual feasible set. Then we show that the dual attainmant in vertices can be ensured under certain additional assumptions.

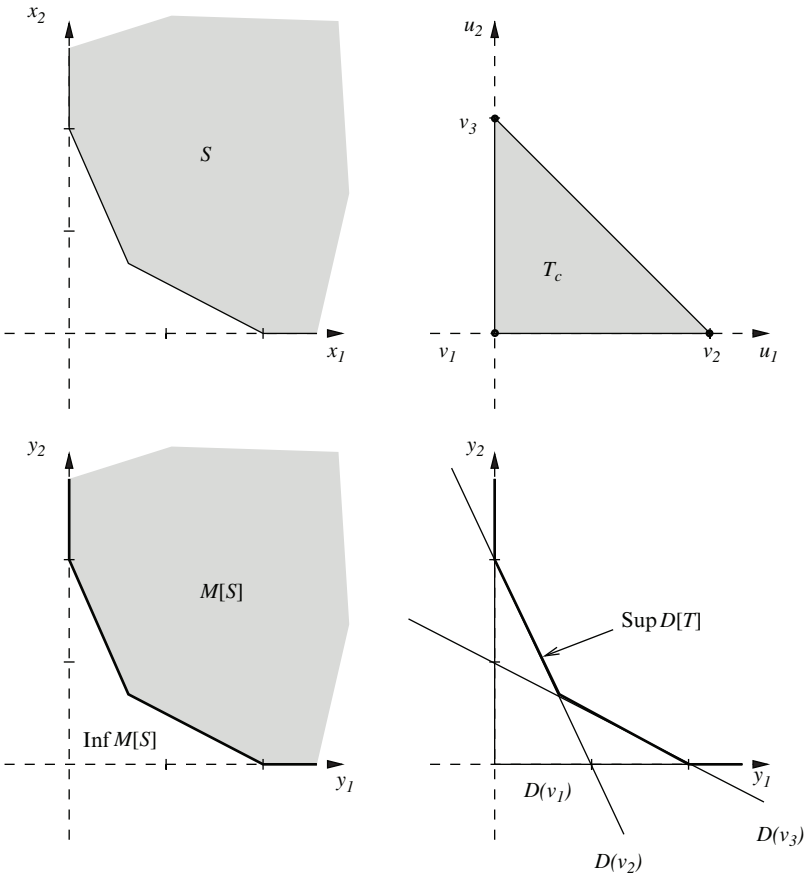


Fig. 1 The primal and dual problem in Example 1

Example 2. Let $q = m = n = 2$, $C = \mathbb{R}^2_+$ and consider the problem (LP^2) with the data

$$M = \begin{pmatrix} \frac{1}{2} & 1 \\ 0 & -1 \end{pmatrix}, \quad A = \begin{pmatrix} -\frac{1}{2} & 1 \\ \frac{3}{2} & -2 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ -2 \end{pmatrix}.$$

As above, we set $c = (1,1)^T \in \operatorname{int}\mathbb{R}^2_+$, hence $B_c = \{c_1^*, c_2^* \geq 0 \,|\, c_1^* + c_2^* = 1\}$. One easily verifies that the dual feasible set is the set $T_c = \operatorname{conv}\{v_1, v_2, v_3, v_4\}$, where $v_1 = (0,0)^T$, $v_2 = (1,0)^T$, $v_3 = (5,2)^T$ and $v_4 = (0,1/3)^T$. However, the four vertices of T_c don't generate the supremum, in fact we have (see Fig. 2)

$$\operatorname{Sup} \bigcup_{i=1}^{4} D_c(v_i) = D_c(v_1) = \operatorname{Inf} M[\mathbb{R}^2_+] = \mathbb{R}_+(0,1)^T \cup \mathbb{R}_+(1,-1)^T.$$

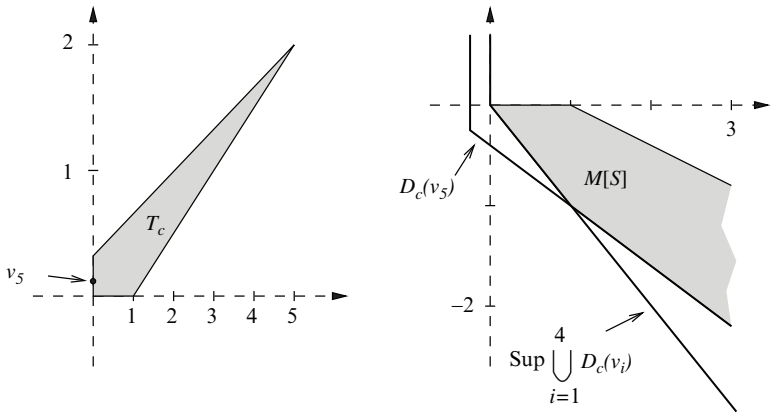


Fig. 2 The dual feasible set and certain values of the dual objective in Example 2

One can show that $v_1 = (0,0)^T$ together with $v_5 = (0,1/8)^T$ generate the supremum, i.e.,

$$\bar{D}_c = \mathrm{Sup}\Big(D_c\big((0,0)^T\big) \cup D_c\big((0,1/8)^T\big)\Big),$$

but v_5 is not a vertex of T_c.

It is natural to ask for additional assumptions to ensure that the supremum of the dual problem is always generated by the vertices (or extreme points) of the dual feasible set T_c. We can give a positive answer for the problem (LD_c^1) by the following considerations. Moreover, we show that problem (LD_c^1) can be simplified under relatively mild assumptions.

Proposition 1. *Let $M \in \mathbb{R}^{q\times n}$, $\mathrm{rank}\, M = q$, $u \in \mathbb{R}^q$, $v \in \mathbb{R}^n$. Then, for the matrix $H := M - uv^T$ it holds $\mathrm{rank}\, H \geq q-1$.*

Proof. We suppose $q \geq 2$, otherwise the assertion is obvious. The matrix consisting of the first k columns $\{a_1, a_2, \ldots, a_k\}$ of a matrix A is denoted by $A^{(k)}$. Without loss of generality we can suppose $\mathrm{rank}\, M^{(q)} = q$. Assume that $\mathrm{rank}\, H^{(q)} =: k \leq q-2$. Without loss of generality we have $\mathrm{rank}\, H^{(k)} = k$. Since $\mathrm{rank}\, H^{(k+1)} = k$, there exist $w \in \mathbb{R}^{k+1} \setminus \{0\}$ such that $H^{(k+1)}w = 0$, hence $M^{(k+1)}w = u(v^T)^{(k+1)}w$. We have $\mathrm{rank}\, M^{(k+1)} = k+1$, hence $(v^T)^{(k+1)}w \neq 0$. It follows that $u \in M^{(k+1)}[\mathbb{R}^{k+1}] = \mathrm{lin}\,\{m_1, m_2, \ldots, m_{k+1}\} =: L$. Thus, for all $x \in \mathbb{R}^{k+1}$, we have $H^{(k+1)}x = M^{(k+1)}x - u(v^T)^{(k+1)}x \in L + L = L$. From $\mathrm{rank}\, M^{(q)} = q$ we conclude that $m_{k+2} \notin L$, hence $h_{k+2} = m_{k+2} + uv_{k+2} \notin L$. It follows that $\mathrm{rank}\, H^{(q)} \geq k+1$, a contradiction. Thus, we have $\mathrm{rank}\, H \geq \mathrm{rank}\, H^{(q)} \geq q-1$. □

Proposition 2. *Let $M \in \mathbb{R}^{q\times n}$, $\mathrm{rank}\, M = q$, $c \in \mathrm{int}\, C$, $c^* \in B_c$ and $H_{c^*} := M - cc^{*T}M$. It holds*

(i) $\operatorname{rank} H_{c^*} = q-1$,
(ii) $H_{c^*}[\mathbb{R}^n]$ *is a hyperplane in* $\mathbb{R}^q$ *orthogonal to* c^*,
(iii) $\operatorname{Inf} H_{c^*}[\mathbb{R}^n] = H_{c^*}[\mathbb{R}^n]$.

Proof. (i) We easily verify that $c^{*T} H_{c^*} = 0$, and so $\operatorname{rank} H_{c^*} < q$. Thus, the statement follows from Proposition 1. (ii) is immediate. (iii) We first show that $H_{c^*}[\mathbb{R}^n] + \operatorname{int} C = \left\{ y \in \mathbb{R}^q \middle| \, c^{*T} y > 0 \right\}$. Of course, for $y \in H_{c^*}[\mathbb{R}^n] + \operatorname{int} C$ we have $c^{*T} y > 0$. Conversely, let $c^{*T} y > 0$ for some $y \in \mathbb{R}^q$. Then, there exists some $\lambda > 0$ such that $c^{*T} \lambda y = 1 = c^{*T} c$. It follows $\lambda y - c \in H_{c^*}[\mathbb{R}^n]$ and hence $y \in H_{c^*}[\mathbb{R}^n] + \operatorname{int} C$.

(a) $H_{c^*}[\mathbb{R}^n] \subseteq \operatorname{Inf} H_{c^*}[\mathbb{R}^n]$. Assume that $y \in H_{c^*}[\mathbb{R}^n]$, but $y \notin \operatorname{Inf} H_{c^*}[\mathbb{R}^n]$. Then by (1) we have $y \in H_{c^*}[\mathbb{R}^n] + \operatorname{int} C$, hence $c^{*T} y > 0$, a contradiction.

(b) $\operatorname{Inf} H_{c^*}[\mathbb{R}^n] \subseteq H_{c^*}[\mathbb{R}^n]$. Let $y \in \operatorname{Inf} H_{c^*}[\mathbb{R}^n]$ and take into account (1). On the one hand this means $y \notin H_{c^*}[\mathbb{R}^n] + \operatorname{int} C$ and hence $c^{*T} y \leq 0$. On the other hand we have $\{y\} + \operatorname{int} C \subseteq H_{c^*}[\mathbb{R}^n] + \operatorname{int} C$, i.e., for all $\lambda > 0$ it holds $c^{*T}(y + \lambda c) > 0$, whence $c^{*T} y \geq 0$. Thus, $c^{*T} y = 0$, i.e., $y \in H_{c^*}[\mathbb{R}^n]$. □

Theorem 2. *Consider problem* $(\mathrm{LD}^1_{\mathrm{c}})$, *where* $M \in \mathbb{R}^{q \times n}$, $\operatorname{rank} M = q$, $c \in \operatorname{int} C$. *Let the matrix* $L \in \mathbb{R}^{q \times m}$ *be defined by* $L := (MM^T)^{-1} M A^T$. *Then it holds*

(i) For $u \in \mathbb{R}^m$, $c^* \in \mathbb{R}^q$: $(A^T u = M^T c^* \quad \Longrightarrow \quad c^* = Lu)$.
(ii) $L[T_c] \subseteq B_c$.
(iii) The dual objective $D_c : T_c \to \mathscr{I}$, $D_c(u) := c\,u^T b \; + \; \operatorname{Inf}(M - c\,u^T A)[\mathbb{R}^n]$ *can be expressed as*

$$D_c(u) = \left\{ y \in \mathbb{R}^q \middle| \; \langle Lu, y \rangle = \langle u, b \rangle \right\}.$$

(iv) For $u_1, u_2, \ldots, u_r \in T_c$, $\lambda_i \geq 0 \;\; (i = 1, \ldots, r)$ *with* $\sum_{i=1}^r \lambda_i = 1$ *it holds*

$$D_c\left(\sum_{i=1}^r \lambda_i u_i \right) \preccurlyeq \operatorname{Sup} \bigcup_{i=1}^r D_c(u_i).$$

(v) If $\bar{D}_c \subseteq \mathbb{R}^q$, *the supremum of* $(\mathrm{LD}^1_{\mathrm{c}})$ *is generated by the set* $\operatorname{ext} T_c$ *of extreme points of* T_c, *i.e.,*

$$\operatorname{Sup} \bigcup_{u \in T_c} D_c(u) = \operatorname{Sup} \bigcup_{u \in \operatorname{ext} T_c} D_c(u).$$

Proof. (i) Since $\operatorname{rank} M = q$, $MM^T \in \mathbb{R}^{q \times q}$ is invertible, and so the statement is easy to verify.

(ii) Let $u \in T_c$. Hence there exists $c^* \in B_c$ such that $A^T u = M^T c^*$. By (i) it follows that $c^* = Lu$, whence $Lu \in B_c$.

(iii) Let $u \in T_c$. By (i) we have $A^T u = M^T Lu$. From Proposition 2 we obtain $D_c(u) = c\,u^T b \; + \; \{ y \in \mathbb{R}^q | \; \langle Lu, y \rangle = 0 \} =: B_1$. Of course, we have $B_1 \subseteq B_2 := \{ y \in \mathbb{R}^q | \; \langle Lu, y \rangle = \langle u, b \rangle \}$. To see the opposite inclusion, let $y \in B_2$, i.e., $\langle Lu, y \rangle = \langle u, b \rangle$. It follows $c\,(Lu)^T y = cu^T b$. By (ii), we have $c^* := Lu \in B_c$. With the aid of Proposition 2, we obtain $y = cu^T b + y - cc^{*T} y \in \left\{ cu^T b \right\} + (I - cc^{*T} I)[\mathbb{R}^n] = B_1$.

(iv) Consider the function $d_c : T_c \to \mathscr{F}$, defined by $d_c(u) := j(D_c(u)) = \mathrm{Cl}_+ D_c(u)$. Proceeding as in the proof of Proposition 2 (iii), we obtain $d_c(u) = \{y \in \mathbb{R}^q |\ \langle Lu, y\rangle \geq \langle u, b\rangle\}$ for all $u \in T_c$. One easily verifies the inclusion $d_c\big(\sum_{i=1}^r \lambda_i u_i\big) \supseteq \bigcap_{i=1}^r d_c(u_i)$. Since j is an isotone bijection between $(\mathscr{I}, \preccurlyeq)$ and $(\mathscr{F}, \supseteq)$, this is equivalent to the desired assertion.

(v) Let $u \in T_c$ be given. Since T_c is closed and convex and contains no lines, [18, Theorem 18.5] yields that there are extreme points $u_1, \ldots, u_k$ and extreme directions $u_{k+1}, \ldots, u_l$ of T_c such that

$$u = \sum_{i=1}^{l} \lambda_i u_i, \quad \text{with} \quad \lambda_i \geq 0 \quad (i = 1, \ldots, l), \quad \text{and} \quad \sum_{i=1}^{k} \lambda_i = 1,$$

(see [18, Sects. 17 and 18]). Of course, we have $v := \sum_{i=k+1}^{l} \lambda_i u_i \in 0^+ T_c$ and $u - v \in T_c$ (where $0^+ T_c$ denotes the asymptotic cone of T_c). From (iv) we obtain

$$D_c(u-v) = D_c\left(\sum_{i=1}^{k} \lambda_i u_i\right) \preccurlyeq \sup_{i=1,\ldots,k} D_c(u_i) \preccurlyeq \sup_{u \in \mathrm{ext}\, T_c} D_c(u).$$

It remains to show that $D_c(u) \preccurlyeq D_c(u-v)$.

Consider the set $V := \{u - v\} + \mathbb{R}_+ v \subseteq T_c$. By (ii), it holds $L[V] = L(u-v) + L[\mathbb{R}_+ v] \subseteq B_c$. Since $L[\mathbb{R}_+ v]$ is a cone, but B_c is bounded it follows that $L[\mathbb{R}_+ v] = \{0\}$. This implies $L(u - v + \lambda v) = c^*$ for all $\lambda \geq 0$, in particular, $L(u-v) = Lu = c^*$. From (iii), we now conclude that exactly one of the following assertions is true:

$$D_c(u) \preccurlyeq D_c(u-v) \quad \text{or} \quad \big(D_c(u-v) \preccurlyeq D_c(u) \ \wedge\ D_c(u-v) \neq D_c(u)\big).$$

We show that the second assertion yields a contradiction. Since $c^* = Lu = L(u-v) \in B_c \subseteq -C^\circ \setminus \{0\}$, we have $\langle u - v, b\rangle < \langle u, b\rangle$ and so $\langle v, b\rangle > 0$ in this case, hence $\langle u - v + \lambda v, b\rangle \to +\infty$ for $\lambda \to +\infty$. It follows that

$$\bar{D}_c = \sup_{u \in T_c} D_c(u) \succcurlyeq \sup_{u \in V} D_c(u) = \sup_{\lambda \geq 0} \{y \in \mathbb{R}^q |\ \langle c^*, y\rangle = \langle u - v + \lambda v, b\rangle\} = \{+\infty\}.$$

This contradicts the assumption $\bar{D}_c \subseteq \mathbb{R}^m$. □

Corollary 1. *Consider problem* $(\mathrm{LD}_{\mathrm{c}}^1)$*, where* $M \in \mathbb{R}^{q \times n}$*,* $\mathrm{rank}\, M = q$*,* $c \in \mathrm{int}\, C$ *and* C *polyhedral.*

If the supremum $\bar{D}_c$ *of* $(\mathrm{LD}_{\mathrm{c}}^1)$ *is a subset of* $\mathbb{R}^q$*, then it is generated by the finitely many vertices* $u_1, \ldots, u_k$ *of (the nonempty polyhedral set)* T_c*. Moreover, we have*

$$\bar{D}_c = \mathrm{Sup} \bigcup_{i=1,\ldots,k} D_c(u_i) = \mathrm{Max} \bigcup_{i=1,\ldots,k} D_c(u_i).$$

Proof. The set T_c is polyhedral [14, Proposition 7.4]. Hence, $\mathrm{ext}\, T_c$ consists of finitely many points, called the vertices of T_c. The first equality follows from Theorem 2(v)

To show the second equality let $y \in \operatorname{Sup}\bigcup_{i=1,\dots,k} D_c(u_i) \subseteq \mathbb{R}^q$ be given. By an assertion analogous to (1) this means $y \notin \bigcup_{i=1,\dots,k} D_c(u_i) - \operatorname{int} C$ and $\{y\} - \operatorname{int} C \subseteq \bigcup_{i=1,\dots,k} D_c(u_i) - \operatorname{int} C$. From the last inclusion we conclude that $y \in \operatorname{cl} \bigcup_{i=1,\dots,k} (D_c(u_i) - \operatorname{int} C) = \bigcup_{i=1,\dots,k} \operatorname{cl}(D_c(u_i) - \operatorname{int} C) = \bigcup_{i=1,\dots,k} (D_c(u_i) - C)$. Hence there exists some $i \in \{1,\dots,k\}$ such that $y \in (D_c(u_i) - C) \setminus (D_c(u_i) - \operatorname{int} C)$. By the same arguments as used in the proof of Proposition 2 (iii) we can show the last statement means $y \in D_c(u_i)$, i.e. we have $y \in \bigcup_{i=1,\dots,k} D_c(u_i)$. The statement now follows from an assertion analogous to (2). □

It is typical that more than one vertex is necessary to generate the infimum or supremum in case of vectorial linear programming. It remains the question how to determine a minimal subset of vertices of S and T_c that generates the infimum and supremum, respectively.

4 Comparison with Duality Based on Scalarization

Duality assertions for linear vector optimization problems are derived by many authors (compare Isermann [10] and Jahn [11]). In these approaches the dual problem is constructed in such a way that the dual variables are linear mappings from $\mathbb{R}^m \to \mathbb{R}^q$, whereas in our approach the dual variables are vectors belonging to $\mathbb{R}^m$. In order to show strong duality assertions these authors suppose that $b \neq 0$. As shown in Theorem 1 we do not need such an assumption in order to prove strong duality assertions. However, there are several relations between our dualtity statements and those given by Jahn [11]. First, we recall an assertion given by Jahn [11] in order to compare our results with corresponding duality statements given by Jahn and others. In the following we consider (LD_c^1) for some $c \in \operatorname{int} C$.

Theorem 3. *(Jahn [11], Theorem 2.3)*
Assume that $\mathbb{V}$ and $\mathbb{Y}$ are real separated locally convex linear spaces and $b \in \mathbb{V}$, $u^* \in \mathbb{V}^*$, $y \in \mathbb{Y}$, $\lambda^* \in \mathbb{Y}^*$.

(i) If there exists a linear mapping $Z : \mathbb{V} \to \mathbb{Y}$ with $y = Z(b)$ and $u^* = Z^*(\lambda^*)$, then $\lambda^*(y) = u^*(b)$.
(ii) If $b \neq 0$, $\lambda^* \neq 0$ and $\lambda^*(y) = u^*(b)$, then there exists a continuous linear mapping $Z : \mathbb{V} \to \mathbb{Y}$ with $y = Z(b)$ and $u^* = Z^*(\lambda^*)$.

Usually, one considers in linear vector optimization dual problems of the form (Isermann [10] and Jahn [11])

$$(\mathrm{LD}^o) \qquad \operatorname{Max} \bigcup_{Z \in T_c^o} Zb,$$

where

$$T_c^o := \{Z \in \mathbb{R}^{q \times m} \mid \exists c^* \in B_c : \; Z^T c^* \geq 0, (ZA)^T c^* = M^T c^*\}. \tag{3}$$

We have the following relationships between (LD_c^1) and (LD^o):

Theorem 4. *Let $M \in \mathbb{R}^{q\times n}$, $\operatorname{rank} M = q$, $c \in \operatorname{int} C$ and $L := (MM^T)^{-1}MA^T \in \mathbb{R}^{q\times m}$. Then it holds*

$$D_1 := \bigcup_{u\in T_c} \{y \in \mathbb{R}^q \mid \langle Lu, y\rangle = \langle u, b\rangle\} \supseteq \bigcup_{Z\in T_c^o} Zb =: D_2$$

In the case of $b \neq 0$ we have equality, i.e., $D_1 = D_2$.

Proof. (a) We show $D_2 \subseteq D_1$. Assume $y \in D_2$. Then there exists $Z \in T_c^o$ with a corresponding $c^* \in B_c$, i.e.,

$$Z^T c^* \geq 0, \quad (ZA)^T c^* = M^T c^* \tag{4}$$

and $y = Zb$. Put $u = Z^T c^*$, then Theorem 3(i) yields

$$\langle c^*, y\rangle = \langle u, b\rangle. \tag{5}$$

Furthermore, taking into account (4) we obtain

$$u = Z^T c^* \geq 0$$

and

$$A^T u = A^T Z^T c^* = (ZA)^T c^* = M^T c^*.$$

By Theorem 2(i) and (5), we conclude that $y \in D_1$.

(b) We show $D_1 \subseteq D_2$ under the assumption $b \neq 0$. Suppose $y \in D_1$. Then there exists $u \in T_c$ with the corresponding $c^* \in B_c$, i.e.

$$u \geq 0, \tag{6}$$

$$A^T u = M^T c^* \tag{7}$$

and

$$\langle Lu, y\rangle = \langle u, b\rangle.$$

From Theorem 2(i), (ii) we get $\langle c^*, y\rangle = \langle u, b\rangle$ and $c^* \in B_c$. So the assumptions $c^* \neq 0$, $b \neq 0$ of Theorem 3(ii) are fulfilled and we conclude that there exists $Z \in \mathbb{R}^{q\times m}$ with $y = Zb$ and $u = Z^T c^*$. Moreover, we obtain by (7)

$$(ZA)^T c^* = A^T Z^T c^* = A^T u = M^T c^*$$

and from (6) we get

$$Z^T c^* = u \geq 0.$$

This yields $y \in D_2$, which completes the proof. □

Remark 1. Theorem 4 shows, that for linear vector optimization problems under the assumption $b \neq 0$ our dual problems coincides with the dual problems given in the papers [10] and [11].

The following example shows that the assumption $b \neq 0$ cannot be omitted in order to have the equality $D_1 = D_2$.

Example 3. Let $q = m = n = 2$, $C = \mathbb{R}^2_+$, $c = (1,1)^T$,

$$A = M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Then we have $L = M = A$, $T_c = B_c$, $D_2 = \left\{(0,0)^T\right\}$ and

$$D_1 = \left\{ y \in \mathbb{R}^2 \middle| \, (y_1 \geq 0 \wedge y_2 \leq 0) \vee (y_1 \leq 0 \wedge y_2 \geq 0) \right\},$$

i.e., $D_1 \neq D_2$.

References

1. Armand P, Malivert C (1991) Determination of the efficient set in multiobjective linear programming. J Optim Theory Appl 70(3):467–489
2. Ecker JG, Hegner NS, Kouada IA (1980) Generating all maximal efficient faces for multiple objective linear programs. J Optim Theory Appl 30:353–381
3. Ecker JG, Kouada IA (1978) Finding all efficient extreme points for multiple objective linear programs. Math Program 14:249–261
4. Ehrgott M, Puerto J, Rodriguez-Chia AM (2006) A primal-dual simplex method for multiobjective linear programming. J Optim Theory Appl 134(3):483–497
5. Evans JP, Steuer RE (1973) A revised simplex method for linear multiple objective programs. Math Program 5:375–377
6. Gal T (1977) A general method for determining the set of all efficient solutions to a linear vectormaximum problem. Eur J Oper Res 1:307–322
7. Hamel A, Heyde F, Löhne A, Tammer Chr, Winkler K (2004) Closing the duality gap in linear vector optimization. J Convex Anal 11(1):163–178
8. Hartley R (1983) Survey of algorithms for vector optimization problems in multi-objective decision making. Multi-objective decision making. In: French S, Hartley R, Thomas LC, and White DJ (eds) Multi-objective decision making. Academic Press, New York
9. Isermann H (1977) The enumeration of the set of all efficient solutions for a linear multiple objective program. Oper Res Quart 28:711–725
10. Isermann H (1978) On some relations between a dual pair of multiple objective linear programs. Z Oper Res Ser A 22:33–41
11. Jahn J (1986) Mathematical vector optimization in Partially ordered linear spaces. Verlag, Peter Lang Frankfurt
12. Löhne A (2005) Optimization with set relations. Dissertation, Martin–Luther–Universität Halle–Wittenberg, Germany
13. Löhne A (2005) Optimization with set relations: conjugate duality. Optimization 54(3):265–282
14. Löhne A, Tammer C (2007) A new approach to duality in vector optimization. Optimization 56(1–2):221–239
15. Nieuwenhuis JW (1980) Supremal points and generalized duality. Math Operationsforsch Stat Ser Optim 11:41–59
16. Philip J (1972) Algorithms for the vector maximization problem. Math Program 2:207–229
17. Philip J (1977) Vector maximization at a degenerate vertex. Math Program 13:357–359
18. Rockafellar RT (1970) Convex analysis. Princeton University Press, Princeton, NJ

19. Rockafellar RT, Wets RJB (1998) Variational analysis. Springer, Berlin
20. Tanino T (1988) On the supremum of a set in a multi-dimensional space. J Math Anal Appl 130:386–397
21. Yu PL, Zeleny M (1975) The set of all nondominated solutions in linear cases and a multicriteria simplex method. J Math Anal Appl 49:430–468
22. Zeleny M (1974) Linear multi-objective programming. Springer, New York

Optimality of the Methods for Approximating the Feasible Criterion Set in the Convex Case

Roman Efremov and Georgy Kamenev

Abstract Estimation Refinement (ER) is an adaptive method for polyhedral approximations of multidimensional convex sets. ER is used in the framework of the Interactive Decision Maps (IDM) technique that provides interactive visualization of the Pareto frontier for convex sets of feasible criteria vectors. We state that, for ER, the number of facets of approximating polytopes is asymptotically multinomial of an optimal order. Furthermore, the number of support function calculations, needed to be resolved during the approximation, and which complexity is unknown beforehand since a user of IDM provides his own optimization algorithm, is bounded from above by a linear function of the number of iterations.

Keywords: Goal programming · Feasible goals method · Interactive decision maps · Estimation refinement

1 Introduction

The Estimation Refinement (ER) method is the first adaptive method for polyhedral approximations of multidimensional convex sets [3]. It is based on computing the support function of the approximated set for certain directions specified adaptively. The method turned out to be an effective tool for approximating the sets of feasible criterion vectors, so-called Feasible Criterion Set (FCS), in the decision problems with convex decision sets and in the case when the number of criteria is not greater than eight. The numerical scheme of the ER method [4], computationally stable to the round-off errors, is implemented in various software, see www.ccas.ru/mmes/mmeda/soft/. The ER-based software is used in the framework of the Interactive Decision Maps (IDM) technique, which provides interactive

R. Efremov (✉)
Rey Juan Carlos University, c/ Tulipán s/n, Móstoles, Madrid, 28933, Spain
e-mail: roman.efremov@urjc.es

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

visualization of the Pareto frontier. In IDM, the maximal set in criterion space, which has the same Pareto frontier as the FCS (Edgeworth-Pareto Hull (EPH) of the FCS), is approximated. Then, the Pareto frontier is visualized interactively by displaying the *decision maps*, that is, collections of superimposed bi-criterion slices of EPH, while several constraints on the value of a "third" criterion are imposed. To present more that three criteria, animation of the decision maps is used.

The IDM technique provides easy and user-friendly interface for exploring the Pareto frontier of FCS. The combination of IDM and the goal-programming approach, so-called Feasible Goals Method (FGM), turned out to be a successful support decision method, see [13]. It consists in a single-shot goal identification on one of a decision map.

The FGM method has been intensively used in various standalone decision support systems, see e.g. [2, 11], as well as in the web-based decision aid tools, see e.g. [8, 14].

In implementations of IDM, EPH is approximated in advance and is separated from the human-computer exploration of decision maps. At the same time, slices of the approximation of EPH can be computed in a moment. This feature of the IDM technique facilitates implementation on computer networks, where decision maps may be depicted and animated on-line. It is based on simple web client-server architecture: the approximation of EPH is accomplished on a server side, while the exploration of the Pareto frontier is carried out by means of Java applets on the user's computer, as explained in [13, 14]. The approximation of EPH requires up to 99% of the computing efforts and can be performed automatically. Having this in a view, we want to be sure beforehand that the approximation will not exceed the time limits and will generally be solved. That is why theoretical as well as experimental study of approximation methods has always been an important task, see [13].

The theoretical study of the ER method gave rise to the concept of *Hausdorff methods* for polyhedral approximation [10]. It was proven in [9] that the optimal order of convergence of approximating polytopes for smooth CCBs equals to $2/(d-1)$ where d is the dimension of the space. It was shown in [10] that Hausdorff methods approximate smooth CCBs with the optimal order of convergence with respect to the number of iterations and vertices of approximating polytopes. Since the ER method belongs to the class of Hausdorff methods [12], it was the first technique, for which it was proven that it has the optimal order of convergence with respect to the number of iterations and vertices. Here we state that, for the Hausdorff methods, the order of the number of facets of approximating polytopes is also optimal. In addition, we state that the order of the number of support function calculations in ER for a class of smooth CCB is optimal, too. The detailed proof of these results is somewhat tedious and does not suit the format of this paper the proof can be found in [7]; here we only provide the results.

The outline of the paper is as follows. In Sect. 2 we describe the FGM/IDM technique and the ER method. We bring an example study of a decision map and discuss important characteristics of ER as an approximation tool in the framework of FGM. Section 3 is devoted to formulation of our results. We obtain these results for the general case first and then adopt them to the ER method. We end up with some discussion.

2 The ER Method in the Framework of FGM/IDM Method

Let us formulate the problem the FGM method solves. Let X be the *feasible decision set* of a problem and $f : X \to \bar{\mathrm{N}}^d$ be a mapping from X to the criterion space $\bar{\mathrm{N}}^d$: the performance of each feasible decision $x \in X$ is described by a d-dimensional vector $y = f(x)$. Here, $Y := f(X)$ is the FCS of the problem. We shall assume Y to be compact. With no loss of generality, we shall assume that the criteria must be maximized. This defines a Pareto order in the criterion space: y dominates y (in the Pareto sense) if, and only if, $y \geq y$ and $y \neq y$. The Pareto frontier of the set Y is defined as $P(Y) := \{y \in Y : \{y \in Y : y \geq y, y \neq y\} = \emptyset\}$. Let $\bar{\mathrm{N}}^d_-$ be the non-positive orthant in $\bar{\mathrm{N}}^d$. The set $H(Y) = Y + \bar{\mathrm{N}}^d_-$ is known as the Edgeworth-Pareto Hull of Y. $H(Y)$ is the maximal set that satisfies $P(H(Y)) = P(Y)$. The FGM method is, though, a multiobjective programming technique that represents the information about the set $P(H(Y))$ through its visualization.

2.1 The IDM Technique

The key feature of IDM consists of displaying the Pareto frontier for more than two criteria through interactive display of bi-criterion slices of $H(Y)$. A bi-criterion slice is defined as follows. Let (y_1, y_2) designates two specified criteria, the so-called "axis" criteria, and z denotes the remaining criteria, which we shall fix at $z^* \in \bar{\mathrm{N}}^{d-2}$. A bi-criterion slice of $H(Y)$, parallel to the criterion plane (y_1, y_2) and related to z^*, is defined as $G(H(Y), z^*) = \{(y_1, y_2) : (y_1, y_2, z^*) \in H(Y)\}$. Note that a slice of $H(Y)$ contains all feasible combinations of values for the specified criteria when the values of the remaining criteria are not worse than z^*. Bi-criterion slices of $H(Y)$ are used in the IDM technique by displaying *decision maps*. To define a particular decision map, the user has to choose a "third", or colour- associated, criterion. Then, a decision map is a collection of superimposed slices, for which the values of the colour-associated criterion change, while the values of the remaining criteria are fixed. Moreover, the slice for the worst value of this criterion encloses the slice for the better one.

An example of a decision map is given on the Fig. 1. Here, a conflict that can take place in an agriculturally developed area is represented. A lake located in the area is used for irrigation purposes. It is also a recreational zone for the residents of a nearby town. The conflict is described by three criteria: agricultural production, level of the lake, and additional water pollution in the lake. In Fig. 1, the trade-off curves, production versus level of the lake, are depicted for several values of pollution. Production is given in the horizontal axis, whereas the lake level is given in vertical axis. The constraints imposed on pollution are specified by the colour scale located under the decision map. Any trade-off curve defines the limits of what can be achieved, say, it is impossible to increase the values of agricultural production and level of the lake beyond the frontier, given a value of the lake pollution. The internal trade-off curve, marked by points C and D, is related to minimal, i.e. zero,

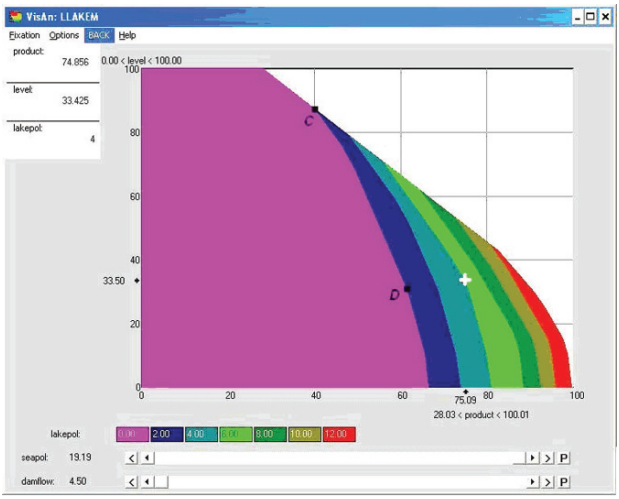


Fig. 1 A decision map

pollution. It shows how the lake level must be decreased to increase production, while keeping the zero level of additional pollution. For small values of the production, about 30%, the maximal level (100%) of the lake is feasible. Then, with the increment in the production, the maximal feasible level of the lake starts to decrease more and more abruptly, especially, after point *C*. The maximal feasible value of the production for zero pollution, a little bit less, than 70%, is related to the minimal level of the lake. Note that it is necessary to exchange a substantial drop of the level (about 30% starting at point *D*) for a small increment in the production needed to achieve its maximal feasible value.

Other trade-off curves have a similar shape. Note that as the level of additional pollution increases, the production level increases as well; nevertheless, if the lake level is reasonably high, the trade-off curves are close to each other, which means that, for these lake levels, even a substantial increment in pollution does not result in economic advantages.

If there are more than three criteria, the user may specify the values of a fourth, fifth and more criteria by applying scroll-bars: sliders of scroll-bars help to specify these values manually see Fig. 1. By moving the sliders of the scroll-bars, the user can study on-line how a variation of these values influences a decision map. The animation of decision maps, which is a display of series of decision maps related to the automatically changing value of, say, the fourth criterion, is possible, too.

Before slices of EPH could be generated the EPH itself should be constructed or approximated in the form of a system of linear inequalities. Approximation method that is actively used in the framework of FGM is the ER method. We outline here the scheme of ER for the case of approximation of convex compact bodies. A simple modification of ER [5] gives an opportunity to approximate the EPH with a desired accuracy.

2.2 The ER Method

Let $\mathscr{C}$ be the class of CCB in the Euclidian space $\bar{\mathrm{N}}^d$ with the scalar product $\langle\cdot,\cdot\rangle$ and the Hausdorff metric $\delta(C_1,C_2) := \max\{\sup\{\rho(x,C_2) : x \in C_1\}, \sup\{\rho(x,C_1) : x \in C_2\}\}$, $C_1, C_2 \in \mathscr{C}$. Let $\mathscr{P}$ be the class of convex polytopes. Let $P \in \mathscr{P}$. Let $C \in \mathscr{C}$ and ∂C be its boundary. Denote the set of the vertices of P by $M^t(P)$ and the set of $d-1$ dimensional facets of P by $M^f(P^n)$. Let $M^f(P^n)$ be given in the form of a system of linear inequalities. Let $\mathscr{P}(C) := \{P \in \mathscr{P} : M^t(P) \subseteq \partial C\}$. For $P \in \mathscr{P}(C)$ let $U(P)$ be the list of outer normals to its facets. Let $g(u,C)$ be the support function of C. Let $\mathrm{conv}\{\cdot\}$ means the convex hull of a set. Let $C \in \mathscr{C}$ be approximated.

The ER method

Prior to the $(n+1)$-th iteration of the method, $M^f(P^n)$ must be constructed. Each facet must be given along with the list of vertices that belong to it. Then, the following steps are carried out.

Step 1:

1. Find $u_n \in U(P^n)$ that solves $\max\{g(u,C) - g(u,P^n) : u \in U(P^n)\}$. If $|g(u_n,C) - g(u_n,P^n)| \le \varepsilon$, then stop the method, otherwise proceed to 2.
2. Select a point $p_n \in \partial C$ such that $\langle u_n, p_n\rangle = g(u_n,C)$.

Step 2: Find $M^f(P^{n+1})$ for $P^{n+1} := \mathrm{conv}\{p_n \cup P^n\}$.

Taking into account an important role of approximation methods in the framework of FGM it is of practical interest to estimate the complexity of the ER method. As for the first step, the optimization problem should be solved and, naturally, nothing can be said concerning complexity in general case. The second step is based on the "beneath-beyond" method [9] of constructing of a convex hull of a polytope and a point. A stable implementation of the method is discussed in [4]. The stability is provided due to the form of the facet description in the ER method, see [4, 5]. Furthermore, this implementation does not generate redundant inequalities, that is, on every iteration n, inequalities that belong to $M^f(P^n)$ and only these inequalities are generated, see [13] for details. Thus, to estimate the real complexity of ER we need only to know the convergence rate of ER with regard to the number of optimization problem to solve, since the optimization problem solving may be time consuming, and with regard to the number of facets, since, virtually, it may be very big, see e.g. [13]. We sketch the results we have obtained for these characteristics in the next section. However, the proofs of these results exceed the scope of this paper.

3 Optimal Estimates of Convergence Rates of Methods for Polyhedral Approximation

To start with, let us introduce some notions and results of the general theory of polyhedral approximation of CCB we will need to formulate our results.

3.1 Hausdorff Methods for Internal Polyhedral Approximation

Consider the *augmentation scheme* [10] that is the general algorithmic scheme for the internal polyhedral approximation of CCB with increasing number of vertices. Let $P^0 \in \mathscr{P}(C)$. Given $P^n \in \mathscr{P}(C)$, the $(n+1)$-th iteration consists of two steps: choose $p_n \in \partial C$ and construct the new polytope $P^{n+1} := \mathrm{conv}\{p_n \cup P^n\}$. The augmentation method for polyhedral approximation of $C \in \mathscr{C}$ is called Hausdorff with constant $\gamma > 0$ if it generates the sequence of polytopes $\{P^n\}_{n=0,1,\ldots}$ such that $\delta(P^n, P^{n+1}) \geq \gamma\delta(P^n, C)$ holds for any $n > 0$. Such a sequence itself is called the $H(\gamma, C)$-*(augmentation) sequence*. Let us denote by $\mathscr{C}^2$ the class of CCB with the two times continuously differentiated boundary and the positive principal curvatures. In [6] it was proven that the following estimate cannot be improved for the class of Hausdorff methods:

$$\limsup_{n\to\infty} \delta(C, P^n)[m^t(P^n)]^{2/(d-1)} \leq \frac{2}{(1-\sqrt{1-\gamma})^2} A(C), \tag{1}$$

where $m^t(P)$ is the number of vertices of P and $A(C)$ is a constant that depends on the body $C \in \mathscr{C}^2$. On the basis of this result and a theorem that describes the facial structure of a convex hull of a polytope and a point, see [9], we have found the similar estimate for the convergence rate with respect to the number of facets.

3.2 Main Results for General Case

Let $k_{\partial C}^{\max}$ and $k_{\partial C}^{\min}$ be the maximal and minimal principal curvatures of a surface ∂C. Then $r_{\partial C} := 1/(k_{\partial C}^{\max})^{1/d-1}$ and $R_{\partial C} := 1/(k_{\partial C}^{\min})^{1/d-1}$ be the radiuses of the internal and external rolling balls for C, which are the balls with maximal (minimal) radius that still can "roll" from within (from the outside of) the surface ∂C, see [1]. Let $C(n,d)$ be the number of combinations from n by d. By $m^f(P)$ denote the number of facets of P. Let $f(n) := m^f(P^{n+1}) - m^f(P^n)$.

Theorem 1. *Let $C \in \mathscr{C}^2$ and let $\{P^n\}_{n=0,1,\ldots}$ be the $H(\gamma, C)$-sequence. Then*

$$\limsup_{n\to\infty} f(n) \leq \bar{f}(C, \gamma),$$

where $\bar{f}(C,\gamma) = C\left(\left(5[1-\sqrt{1-\gamma}]^{-1} R_{\partial C}/r_{\partial C}\right)^{d-1}, d-1\right)$.

Theorem 1 and (1) result in the following estimate for the order of convergence of $H(\gamma, C)$- sequences with respect to the number of facets:

Corollary 1.

$$\limsup_{n\to\infty} \delta(C, P^n)[m^f(P^n)]^{2/(d-1)} \leq \frac{2\bar{f}(C,\gamma)}{(1-\sqrt{1-\gamma})^2} A(C),$$

where $\bar{f}(C,\gamma)$ is the same as in Theorem 1.

As can be seen from Corollary 1 and (1), the order of convergence of $H(\gamma, C)$-sequences with respect to the number of facets is the same as with respect to the number of vertices and is optimal.

3.3 Applications to ER

It is shown in [12] that the ER method is Hausdorff; namely, if $\{P_n\}_{n=0,1,\ldots}$ is generated for $C \in \mathscr{C}^2$ by ER, there exists such $N \geq 0$ that $\{P_n\}_{n=N,N+1,\ldots}$ is $H(1-\varepsilon, C)$-augmentation sequence for any $\varepsilon, 0 < \varepsilon < 1$. From this fact and the Theorem 1 follows:

Theorem 2. *Let $\{P^n\}_{n=0,1,\ldots}$ be generated by ER for $C \in \mathscr{C}^2$. Then, it holds:*

$$\limsup_{n\to\infty} \delta(C, P^n)[m^f(P^n)]^{2/(d-1)} \leq 2\bar{f}(C,\gamma)A(C),$$

where $\bar{f}(C,\gamma) = C\left((5R_{\partial C}/r_{\partial C})^{d-1}, d-1\right)$.

Denote by $s(n)$ the number of computations of the support function at the iterations up to n.

Theorem 3. *Let $\{Pn\}_{n=0,1,\ldots}$ be generated by the ER method for $C \in \mathscr{C}^2$. Then, it holds:*

$$\limsup_{n\to\infty} s(n) \leq m^f(P^0) + \bar{f}(C,\gamma)n,$$

where $\bar{f}(C,\gamma)$ is the same as in Theorem 2.

Thus, the value of $s(n)$ is bounded from above by a linear function of n. It follows that its order of convergence is optimal.

4 Conclusions

It was shown theoretically that the order of convergence of the Hausdorff methods for internal polyhedral approximation and compact convex bodies from C 2 are optimal with respect to the number of facets. For the ER method, it was additionally

shown that the number of support function calculations up to the iteration n is bounded from above by a linear function of n. It is important for its applications in the framework of the IDM technique, when approximating EPH, since every support function calculation may be time-consuming, as well as when constructing (animating) decision maps, since the complexity of the algorithm for constructing bi-criterion slices is directly proportional to the number of facets in the approximation of EPH.

Acknowledgements Roman Efremovs research was supported by the Edemocracia-CM project of the University department of Madrid Region and TSI2004- 06801-C04-01 project of the Ministry of Education and Science (MEC) of Spain. Georgy Kamenevs research was supported by the Russian Foundation for Basic Research project no. 07-01-00472, by the grant no. NSh-5379.2006.1 of the President of Russia for State Support of Leading Scientific Schools, by the Program no. 14 for Fundamental Research of the Russian Academy of Sciences, by the Program no. 3 for Fundamental Research of the Department of Mathematical Sciences of RAS. The authors are grateful to Prof. A.Lotov for valuable comments to the paper.

References

1. Brooks LN, Strantzen JB (1989) Blaschke's rolling theorem in $\mathbb{R}^n$. Mem Am Math Soc 80(405):2–5
2. Burmistrova LV, Efremov RV, Lotov AV (2002) A decision-making visual support technique and its application in water resources management systems. J Comput Syst Sci Int 41(5):759–769
3. Bushenkov VA, Lotov AV (1982) Methods for the construction and application of generalized reachable sets. Technical report, Computing Center of the USSR Academy of Sciences, Moscow (In Russian)
4. Chernykh OL (1988) Construction of the convex hull of a finite set of points when the computations are approximate. Comput Math Math Phys 28(5):71–77
5. Chernykh OL (1995) Approximation of the Pareto-hull of a convex set by polyhedral sets. Comput Math Math Phys 35(8):1033–1039
6. Efremov R, Rios Insua D, Lotov A (2006) A framework for participatory group decision support over the web based on Pareto frontier visualization, goal identification and arbitration. Technical reports on statistics and decision sciences, Rey Juan Carlos University, Madrid, Spain
7. Efremov RV, Kamenev GK (2002) An a priori estimate for the asymptotic efficiency of a class of algorithms for the polyhedral approximation of convex bodies. Comput Math Math Phys 42(1):20–29
8. Efremov RV, Kamenev GK (2008) Optimal estimate of the growth order of the number of facets in one class of the methods for polyhedral approximation. Comput Math Math Phys
9. Kamenev GK (1992) A class of adaptive algorithms for the approximation of bodies by polyhedra. Comput Math Math Phys 32(1):114–127
10. Kamenev GK (1994) Investigation of an algorithm for the approximation of convex bodies. Comput Math Math Phys 34(4):521–528
11. Lotov A, Kistanov A, Zaitsev A (2004) Visualization-based data mining tool and its web application. In: Shi Y, Xu W, Chen Z (eds) Data mining and knowledge management, LNAI vol 3327. Springer, Berlin, Heidelberg, pp 1–10

12. Lotov AV, Bourmistrova LV, Efremov RV, Bushenkov VA, Buber AL, Brainin NA (2005) Experience of model integration and Pareto frontier visualization in the search for preferable water quality strategies. Environ Model Software 20(2):243–260
13. Lotov AV, Bushenkov VA, Kamenev GK (2004) Interactive Decision Maps. Approximation and Visualization of Pareto Frontier. Kluwer, Boston
14. Preparata FP, Shamos MI (1985) Computational geometry: an introduction. Spinger, New York

Introducing Nonpolyhedral Cones to Multiobjective Programming

Alexander Engau and Margaret M. Wiecek

Abstract The nondominated set of a multiobjective program is investigated with respect to a class of nonpolyhedral cones, that are defined in direct generalization of Pareto, polyhedral, second order and general p-th order cones. Properties of these cones are derived using the concept of positively homogeneous functions, and two approaches to generating the associated nondominated points are presented. In particular, it is shown how a well known relationship between the nondominated points with respect to a polyhedral cone and Pareto points can be generalized for a nonpolyhedral cone. In addition, several scalarization methods that have originally been formulated for finding Pareto points can be modified to also allow for a general (polyhedral or nonpolyhedral) cone. The results are illustrated on examples and discussed for a specific class of nonpolyhedral cones.

Keywords: Conic scalarization · Domination cones · Multiobjective programming · Positively homogeneous functions

1 Introduction

Let $\mathbb{R}^m$ be a Euclidean space, $Y \subseteq \mathbb{R}^m$ be a nonempty subset, and $C \subseteq \mathbb{R}^m$ be a nonempty cone. The set of nondominated points of Y with respect to the cone C is defined by [21]

$$\mathrm{N}(Y,C) := \{y \in Y : Y \cap (y - C) \subseteq \{y\}\}.$$

The theory of vector (or multicriteria) optimization deals with the characterization of this (and other) sets of nondominated points [3, 12] and with developing multiobjective programming methods for finding or approximating these sets [4,14].

A. Engau (✉)
Department of Management Sciences, University of Waterloo, Canada
e-mail: aengau@alumni.clemson.edu

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

As the references show, the underlying cone C is usually assumed to be Pareto or, slightly more general, polyhedral. Some authors have also studied cases where C is not a cone, but a more general domination structure [20, 22], and derived scalarization methods to characterize the corresponding nondominated points [8].

This paper investigates the case where C is chosen as a nonpolyhedral cone that is defined in direct generalization of Pareto, polyhedral, second order and general p-th order cones. Properties of these cones are derived using the concept of positively homogeneous functions that can be sublinear or superlinear.

The literature reports on various applications of especially second order cones in engineering and robust optimization [13], including portfolio optimization, signal processing, truss topology design, structural engineering, quadratic and robust linear programming, and norm minimization problems [6]. However, the authors of this paper are not aware of any applications of nonpolyhedral cones in multiobjective programming and, therefore, present this research to encourage and propose their further investigation.

This paper studies the set of nondominated points with respect to nonpolyhedral cones and develops approaches for its generation. In particular, it is shown how a well known result relating nondominated points with respect to polyhedral cones and Pareto points [2, 10, 15, 19, 22] can be generalized to nonpolyhedral cones, so that the problem of finding nondominated points with respect to a nonpolyhedral cone can be reduced to finding Pareto points. In addition, several scalarization methods that have originally been formulated for finding Pareto points to multiobjective programs can be modified to also allow for a general (polyhedral or nonpolyhedral) cone. The results are illustrated on examples and discussed for a new class of nonpolyhedral cones.

2 Terminology and Definitions

A *cone* $C \subseteq \mathbb{R}^m$ is a nonempty set for which $c \in C \Rightarrow \lambda c \in C$ whenever $\lambda > 0$. It is said to be *convex* if $c, d \in C \Rightarrow c + d \in C$, and *pointed* if $\sum_{i=1}^{k} c^i = 0 \Rightarrow c^i = 0$ for all $i = 1, \ldots, k$, where the $c^i \in C$ are any k elements of C. In addition, given the cone C, its *dual cone* is defined by $C^+ := \{w \in \mathbb{R}^m : w^T c \geq 0$ for all $c \in C\}$ with interior $\operatorname{int} C^+ = \{w \in \mathbb{R}^m : w^T c > 0$ for all $c \in C \setminus \{0\}\}$.

Definition 1 (Pareto Cone). The m-dimensional Pareto cone $\mathbb{R}^m_{\geq} \subseteq \mathbb{R}^m$ is defined by

$$\mathbb{R}^m_{\geq} := \{c \in \mathbb{R}^m : c \geq 0\}.$$

Definition 2 (Polyhedral Cone). For a matrix $A \in \mathbb{R}^{l \times m}$, the polyhedral cone $C(A) \subseteq \mathbb{R}^m$ associated with A is defined by

$$C(A) := \{c \in \mathbb{R}^m : Ac \geq 0\}.$$

In particular, for $A = I^m \in \mathbb{R}^{m \times m}$ the m-dimensional identity matrix, the polyhedral cone $C(I^m) = \mathbb{R}^m_{\geq}$ is the Pareto cone.

Definition 3 (*p*-th Order Cone). Given $p \geq 1$, the m-dimensional p-th order cone $C_p^m \subseteq \mathbb{R}^m$ is defined by

$$C_p^m := \{c = (c_1, c_{-1}) \in \mathbb{R}^{1+(m-1)} : c_1 \geq \|c_{-1}\|_p\}.$$

In particular, for $p = 2$, the second order cones C_2^m are also called *Lorentz* or *ice cream cones*, and for all $p \geq 1$, the cones C_p^m belong to the more general class of *Bishop–Phelps cones* [11, 17].

A function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ is said to be *positively homogeneous* if $\Gamma(\lambda c) = \lambda\Gamma(c)$ whenever $\lambda > 0$. If, in addition, $\Gamma(c+d) \leq \Gamma(c) + \Gamma(d)$ for all $c, d \in \mathbb{R}^m$, then Γ is said to be *sublinear*. If, instead, $\Gamma(c+d) \geq \Gamma(c) + \Gamma(d)$ for all $c, d \in \mathbb{R}^m$, then Γ is said to be *superlinear*. Finally, if, for all $c, d \in \mathbb{R}^m$, $\Gamma(c) = \Gamma(d)$ if and only if $c = d$, then Γ is said to be *injective*.

Definition 4 (Nonpolyhedral Cone). For a positively homogeneous function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$, the nonpolyhedral cone $C(\Gamma) \subseteq \mathbb{R}^m$ associated with Γ is defined by

$$C(\Gamma) := \{c \in \mathbb{R}^m : \Gamma(c) \geq 0\}.$$

It is clear that $C(\Gamma)$ defines a cone. In particular, the three cones associated with the positively homogeneous functions $\Gamma^1(c) = c$, $\Gamma^2(c) = Ac$, and $\Gamma^3(c) = c_1 - \|c_{-1}\|_p$ are $C(\Gamma^1) = \mathbb{R}^m_{\geq}$, $C(\Gamma^2) = C(A)$, and $C(\Gamma^3) = C_p^m$, respectively.

3 Preliminaries

A positively homogeneous function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ can also be defined under the equivalent conditions that $\Gamma(\lambda c) \geq \lambda\Gamma(c)$, or $\Gamma(\lambda c) \leq \lambda\Gamma(c)$, whenever $\lambda > 0$. This follows, because $\Gamma(\lambda c) \geq \lambda\Gamma(c) = \lambda\Gamma(\lambda^{-1}\lambda c) \geq \lambda\lambda^{-1}\Gamma(\lambda c) = \Gamma(\lambda c) \Rightarrow \Gamma(\lambda c) = \lambda\Gamma(c)$. In particular, $\Gamma(0) = \Gamma(2 \cdot 0) = 2\Gamma(0) \Rightarrow \Gamma(0) = 0$, and thus, $\Gamma(\lambda c) = \lambda\Gamma(c)$ whenever $\lambda \geq 0$.

Proposition 1 (Rockafellar and Wets [18]). *A function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ is sublinear (superlinear) if and only if Γ is positively homogeneous and convex (concave). Moreover, then $\Gamma\left(\sum_{i=1}^k c^i\right) \leq (\geq) \sum_{i=1}^k \Gamma(c^i)$.*

Clearly, if a function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ is sublinear, then $-\Gamma$ is superlinear and vice versa.

Proposition 2 (Polyhedral Cone). *For a matrix $A \in \mathbb{R}^{l \times m}$, let $C(A) \subseteq \mathbb{R}^m$ be the associated polyhedral cone. $C(A)$ is nonempty because $0 \in C(A)$, and always convex. $C(A)$ is pointed if and only if $c \mapsto Ac$ is injective.*

For the pointedness of a polyhedral cone $C(A) \subseteq \mathbb{R}^m$, equivalent conditions are that $\operatorname{rank} A = m \geq 2$ or, since the mapping $c \mapsto Ac$ is linear, that $Ac = 0$ if and only if $c = 0$. In particular, for $A = I^m \in \mathbb{R}^{m \times m}$, the Pareto cone $C(I^m) = \mathbb{R}^m_{\geq}$ is convex and pointed.

Proposition 3 (Nonpolyhedral Cone). *For a positively homogeneous function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$, let $C(\Gamma) \subseteq \mathbb{R}^m$ be the associated nonpolyhedral cone. $C(\Gamma)$ is nonempty because $0 \in C(\Gamma)$, and convex if Γ is superlinear. $C(\Gamma)$ is pointed if, in addition, Γ is injective.*

Proof. Let $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ be positively homogeneous, then $\Gamma(0) = 0$ and, thus, $0 \in C(\Gamma)$. Now let $c, d \in C(\Gamma)$, so $\Gamma(c), \Gamma(d) \geq 0$. If Γ is superlinear, then $\Gamma(c+d) \geq \Gamma(c) + \Gamma(d) \geq 0$ and, thus, $c+d \in C(\Gamma)$, showing that $C(\Gamma)$ is convex. If, in addition, Γ is injective, $\Gamma(c) = 0 \Leftrightarrow c = 0$, let $\sum_{i=1}^k c^i = 0$ with $c^i \in C$, so $\Gamma(c^i) \geq 0$ for all $i = 1, \ldots, k$. Then, by superlinearity,

$$0 = \Gamma(0) = \Gamma\left(\sum_{i=1}^k c^i\right) \geq \sum_{i=1}^k \Gamma(c^i) \geq 0 \Longrightarrow \Gamma(c^i) = 0$$

and, by injectivity, all $c^i = 0$, showing that $C(\Gamma)$ is pointed. □

For the pointedness of a nonpolyhedral cone $C(\Gamma) \subseteq \mathbb{R}^m$, the proof of Proposition 3 only requires that $\Gamma(c) = 0$ if and only if $c = 0$, and thus, if Γ is nonlinear, injectivity of Γ is sufficient, but not necessary. In particular, the superlinear function $\Gamma(c) = c_1 - \|c_{-1}\|_p$ is not injective (e.g., $\Gamma(c) = 0$ for all $c = (c_1, c_{-1})$ with $c_1 = \|c_{-1}\|_p$), but the p-th order cone $C(\Gamma) = C_p^m$ is convex and pointed.

Proposition 4. *If $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ is sublinear, then $-C(-\Gamma) \subseteq C(\Gamma)$. The reverse inclusion holds if, instead, Γ is superlinear. Furthermore, both inclusions become equality if Γ is linear.*

Proof. Let $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$ be sublinear and $c \in -C(-\Gamma)$, so $-\Gamma(-c) \geq 0$. Then $0 = \Gamma(0) = \Gamma(c-c) \leq \Gamma(c) + \Gamma(-c)$ and, thus, $\Gamma(c) \geq -\Gamma(-c) \geq 0$, showing that $c \in C(\Gamma)$. If, instead, Γ is superlinear, then $-\Gamma$ is sublinear and, thus, $-C(\Gamma) \subseteq C(-\Gamma)$, or $C(\Gamma) \subseteq -C(-\Gamma)$. □

4 Nondominated Points and Pareto Points

The following result is well established throughout the literature, see [2, 10, 15, 19, 22], among others.

Theorem 1 (Polyhedral Cone). *For a matrix $A \in \mathbb{R}^{l \times m}$, let $C(A) \subseteq \mathbb{R}^m$ be the associated polyhedral cone. Then*

$$A[\mathrm{N}(Y, C(A)] \subseteq \mathrm{N}(A[Y], \mathbb{R}^l_{\geq}).$$

Furthermore, the above inclusion becomes set equality, if the polyhedral cone $C(A)$ is pointed.

Hence, the problem of finding the nondominated set of Y with respect to a pointed polyhedral cone $C(A)$ is equivalent to finding the Pareto set of $A[Y]$. Corresponding results can be derived for general nonpolyhedral cones $C(\Gamma)$.

Theorem 2 (Nonpolyhedral Cone). *For a positively homogeneous function $\Gamma : \mathbb{R}^m \to \mathbb{R}^l$, let $C(\Gamma) \subseteq \mathbb{R}^m$ be the associated nonpolyhedral cone. If Γ is sublinear, then*

$$\Gamma[\mathrm{N}(Y, C(\Gamma))] \subseteq \mathrm{N}(\Gamma[Y], \mathbb{R}^l_{\geq}).$$

The reverse inclusion holds if, instead, Γ is superlinear and injective. Furthermore, both inclusions become equality if Γ is linear and injective.

Proof. Let $\hat{z} \in \Gamma[\mathrm{N}(Y, C(\Gamma))]$, so $\hat{z} = \Gamma(\hat{y})$ with $\hat{y} \in \mathrm{N}(Y, C(\Gamma))$. Then $Y \cap (\hat{y} - C(\Gamma)) \subseteq \{\hat{y}\}$, and hence, there does not exist $y \in Y \setminus \{\hat{y}\}$ such that $\hat{y} - y \in C(\Gamma)$, or $\Gamma(\hat{y} - y) \geq 0$. Now suppose by contradiction that $\hat{z} \notin \mathrm{N}(\Gamma[Y], \mathbb{R}^l_{\geq})$, then there exists $z = \Gamma(y)$, $z \neq \hat{z}$, so $y \neq \hat{y}$, with $z \in \Gamma(Y) \cap (\hat{z} - \mathbb{R}^l_{\geq})$, thus $\hat{z} - z \in \mathbb{R}^l_{\geq}$, or $\Gamma(\hat{y}) - \Gamma(y) \geq 0$. However, by sublinearity of Γ,

$$\Gamma(\hat{y} - y) + \Gamma(y) \geq \Gamma(\hat{y}) \Longrightarrow \Gamma(\hat{y} - y) \geq \Gamma(\hat{y}) - \Gamma(y) \geq 0,$$

in contradiction to the non-existence of $y \in Y \setminus \{\hat{y}\}$ with $\Gamma(\hat{y} - y) \geq 0$. ↯

For the second statement and the reverse inclusion, let $\hat{z} = \Gamma(\hat{y}) \in \mathrm{N}(\Gamma[Y], \mathbb{R}^l_{\geq})$. Then $\Gamma[Y] \cap (\hat{z} - \mathbb{R}^l_{\geq}) \subseteq \{\hat{z}\}$, and hence, there does not exist $z \in \Gamma[Y] \setminus \{\hat{z}\}$ such that $\hat{z} - z \in \mathbb{R}^l_{\geq}$, or $\hat{z} - z \geq 0$. Since Γ is injective, $\Gamma(y) \neq \Gamma(\hat{y})$ if and only if $y \neq \hat{y}$, and so there does not exist $y \in Y \setminus \{\hat{y}\}$ such that $\Gamma(\hat{y}) - \Gamma(y) \geq 0$. Now suppose by contradiction that $\hat{z} = \Gamma(\hat{y}) \notin \Gamma[\mathrm{N}(Y, C(\Gamma))]$, or $\hat{y} \notin \mathrm{N}(Y, C(\Gamma))$. Then there exists $y \neq \hat{y}$ with $y \in Y \cap (\hat{y} - C(\Gamma))$, thus $\hat{y} - y \in C(\Gamma)$, or $\Gamma(\hat{y} - y) \geq 0$. However, by superlinearity of Γ,

$$\Gamma(\hat{y}) \geq \Gamma(\hat{y} - y) + \Gamma(y) \Longrightarrow \Gamma(\hat{y}) - \Gamma(y) \geq \Gamma(\hat{y} - y) \geq 0,$$

in contradiction to the nonexistence of $y \in Y \setminus \{\hat{y}\}$ with $\Gamma(\hat{y}) - \Gamma(y) \geq 0$. ↯

If Γ is a linear function, then $\Gamma(c) = Ac$ for some matrix $A \in \mathbb{R}^{l \times m}$, and Theorem 2 reduces to Theorem 1 with the polyhedral cone $C(\Gamma) = C(A)$. □

If $C(\Gamma) \subseteq \mathbb{R}^m$ is not polyhedral, then, in general, equality in Theorem 2 cannot be expected.

5 Generating Methods

Several scalarization methods that have originally been formulated for finding Pareto points to multiobjective programs can be modified to also allow for a general (polyhedral or nonpolyhedral) cone C. The original methods and additional references are discussed in [4].

Proposition 5 (Geoffrion Method [7]). *Let $w \in C^+ \setminus \{0\}$, and $\hat{y} \in Y$ be an optimal solution to*

$$\text{minimize } w^T y \text{ subject to } y \in Y. \tag{GeM}$$

(1) If $w \in \operatorname{int} C^+$, or (2) if the optimal solution $\hat{y}$ is unique, then $\hat{y} \in \mathrm{N}(Y,C)$.

Proposition 5 is similarly given in [19], among others. If the cone C is convex, the method can be extended by an additional reference point $y^\circ \in Y$.

Proposition 6 (Guddat Method [9]). *Let C be convex, $w \in C^+ \setminus \{0\}$, $y^\circ \in Y$, and $\hat{y} \in Y$ be an optimal solution to*

$$\text{minimize } w^T y \text{ subject to } y^\circ - y \in C \text{ and } y \in Y. \tag{GuM}$$

(1)If $w \in \operatorname{int} C^+$, or (2) if the optimal solution $\hat{y}$ is unique, then $\hat{y} \in \mathrm{N}(Y,C)$.

Proof. Let $\hat{y} \in Y$ be an optimal solution to (GuM) and, by contradiction, assume that $\hat{y} \notin \mathrm{N}(Y,C)$. Then there exists $y \in Y \cap (\hat{y} - C)$, $y \neq \hat{y}$, so $\hat{y} = y + c$ for some $c \in C \setminus \{0\}$. The point y is feasible for (GuM), because $y^\circ - y = y^\circ - \hat{y} + c \in C$ by feasibility of $\hat{y}$ for (GuM) and convexity of C, but

$$w^T \hat{y} = w^T (y + c) = w^T y + w^T c \begin{cases} > w^T y & \text{in case (i)} \\ \geq w^T y & \text{in case (ii)} \end{cases}$$

in contradiction to the (in case (ii): unique) optimality of $\hat{y}$ to (GeM). □

If $w = e = (1, \ldots, 1)^T \in C^+$ is the unit vector of all ones, then the above method reduces to the classic method proposed by Benson.

Proposition 7 (Benson Method [1]). *Let C be convex, $e \in C^+$, $y^\circ \in Y$, and $\hat{c} \in C$ be an optimal solution to*

$$\text{maximize } \sum_{i=1}^{m} c_i \text{ subject to } y^\circ - c = y \in Y \text{and } c \in C. \tag{BM}$$

(1)If $e \in \operatorname{int} C^+$, or (2) if the solution $\hat{c}$ is unique, then $\hat{y} = y^\circ - \hat{c} \in \mathrm{N}(Y,C)$.

Proof. Since the constraints $y^\circ - c = y \in Y$ and $c \in C$ can be rewritten as $y^\circ - y = c \in C$ and $y \in Y$, and because the maximization of $\sum_{i=1}^m c_i = \sum_{i=1}^m (y_i^\circ - y_i) = \sum_{i=1}^m y_i^\circ - \sum_{i=1}^m y_i$ is equivalent to the minimization of $\sum_{i=1}^m y_i$, Proposition 7 follows from Proposition 6 with $w = e$. □

Only the following method has originally been formulated for a closed convex cone C and can, in particular, be derived as a special case of the results in [8]. The proof is, therefore, omitted.

Proposition 8 (Pascoletti–Serafini Method [16]). *Let $r \in \mathbb{R}^m$, $v \in \operatorname{int} C$, and $\hat{\mu} \in \mathbb{R}$ be an optimal solution to*

$$\text{minimize } \mu \text{ subject to } r + \mu v - c = y \in Y,\ c \in C \text{ and } \mu \in \mathbb{R}. \tag{PSM}$$

Then there exists $\hat{c} \in C$ such that $\hat{y} = r + \hat{\mu} v - \hat{c} \in \mathrm{N}(Y, C)$.

6 Bicriteria Examples

The two theorems in Sect. 4 and the generating methods in Sect. 5 are illustrated for the bicriteria program $\mathrm{N}(Y,C)$ on the unit disk $Y = \{y \in \mathbb{R}^2 : y_1^2 + y_2^2 \leq 1\}$ for three choices of $C \subseteq \mathbb{R}^2$, as shown in Fig. 1. The boundary of Y is the unit circle and denoted by $\partial Y = \{y \in \mathbb{R}^2 : y_1^2 + y_2^2 = 1\}$.

Example 1 (Pareto Cone). Let $\Gamma^1(c) = c$ be the identity, so $C(\Gamma^1) = C(I^2) = \mathbb{R}^2_{\geq}$ be the Pareto cone. Then

$$\Gamma^1[\mathrm{N}(Y, C(\Gamma^1))] = \mathrm{N}(Y, C(I^2)) = \mathrm{N}(Y, \mathbb{R}^2_{\geq}),$$

and Theorems 1 and 2 apply trivially. In particular, (see Fig. 2 (left))

$$\mathrm{N}(Y, C(\Gamma^1)) = \mathrm{N}(Y, \mathbb{R}^2_{\geq}) = \{y \in \partial Y : y \leq 0\}.$$

Furthermore, since the Pareto cone is convex, self-dual (i.e., $C^+(\Gamma^1) = C(\Gamma^1) = \mathbb{R}^2_{\geq}$) and, in particular, $e \in \operatorname{int} C^+ \setminus \{0\} = \operatorname{int} \mathbb{R}^2_{\geq}$, Propositions 5, 6 and 7 apply for (GeM), (GuM) and (BM) in their original formulation.

Example 2 (Polyhedral Cone). Let $\Gamma^2(c) = Ac$ with $A = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) \in \mathbb{R}^{2\times 2}$ be a linear function, so $C(\Gamma^2) = C(A)$ be the associated polyhedral cone. Since rank $A = 2$, $C(A)$ is pointed and Γ^2 is injective, so that, in particular, $A^{-1} = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 1 \end{smallmatrix}\right)$ exists. Then both Theorems 1 and 2 apply and give that

$$\Gamma^2[\mathrm{N}(Y, C(\Gamma^2))] = A[\mathrm{N}(Y, C(A))] = \mathrm{N}(A[Y], \mathbb{R}^2_{\geq}),$$

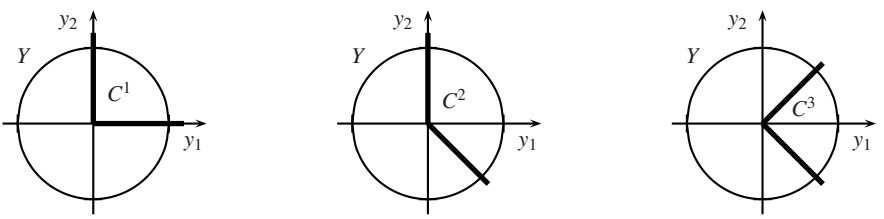


Fig. 1 Pareto, polyhedral and p-th order cone for Examples 1–3 (from left to right)

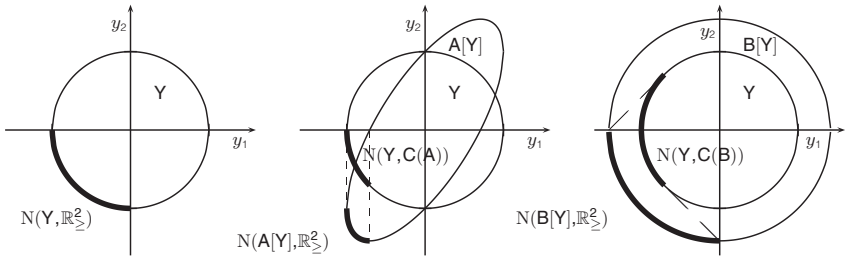


Fig. 2 The sets $\mathrm{N}(Y,C)$ and $\mathrm{N}(\Gamma[Y],\mathbb{R}^2_{\geq})$ for Examples 1–3 (from left to right)

and thus, (see Fig. 2 (center))

$$\mathrm{N}(Y,C(\Gamma^2)) = A^{-1}[\mathrm{N}(A[Y],\mathbb{R}^2_{\geq})] = \{y \in \partial Y : y_2 \leq 0 \text{ and } y_1 - y_2 \leq 0\}.$$

Since $C(A)$ is convex and has the dual cone $C^+(A) = C(A^+)$ with $A^+ = \left(\begin{smallmatrix} -1 & 1 \\ 0 & 1 \end{smallmatrix}\right) \in \mathbb{R}^{2\times 2}$, Propositions 5 and 6 apply for (GeM) and (GuM) with $w \in C^+(\Gamma^2) \setminus \{0\} = C(A^+) \setminus \{0\}$. Since $A^+e = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right) \not> 0$, only (ii) of Proposition 7 applies for (BM). In particular, the Guddat method (GuM) becomes

$$\text{minimize } w^T y \text{ subject to } A(y^\circ - y) \geq 0 \text{ and } y \in Y$$

and, thus, can be solved as a linear program in the objective space.

Example 3 (*p*-th Order Cone). Let $\Gamma^3(c) = c_1 - \|c_2\|_p = c_1 - |c_2|$ be a superlinear function, so $C(\Gamma^3) = C^2_p$ be the associated p-th order cone. Since Γ^3 is not injective, both Theorems 1 and 2 do not apply to Γ^3. However, since $c_1 - |c_2| \geq 0$ can equivalently be written as $c_1 - c_2 \geq 0$ and $c_1 + c_2 \geq 0$, let $\Gamma^4(c) = Bc$ with $B = \left(\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right) \in \mathbb{R}^{2\times 2}$ be a linear function, so $C(\Gamma^3) = C(\Gamma^4) = C(B)$. Since $\operatorname{rank} B = 2$, $C(B)$ is pointed and Γ^4 is injective, so that, in particular, $B^{-1} = \frac{1}{2}\left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right)$ exists. Then Theorems 1 and 2 give that

$$\Gamma^4[\mathrm{N}(Y,C(\Gamma^4))] = B[\mathrm{N}(Y,C(B))] = \mathrm{N}(B[Y],\mathbb{R}^2_{\geq}),$$

and thus, (see Fig. 2 (right))

$$\begin{aligned}\mathrm{N}(Y,C(\Gamma^3)) = \mathrm{N}(Y,C(\Gamma^4)) &= B^{-1}[\mathrm{N}(B[Y],\mathbb{R}^2_{\geq})] \\ &= \{y \in \partial Y : y_1 + y_2 \leq 0 \text{ and } y_1 - y_2 \leq 0\}.\end{aligned}$$

Furthermore, since $C(B)$ is convex and self-dual, $C^+(B) = C(B)$, Propositions 5 and 6 apply for (GeM) and (GuM) with $w \in C^+(\Gamma^3) \setminus \{0\} = C^+(B) \setminus \{0\} = C(B) \setminus \{0\}$. Since $Be = \left(\begin{smallmatrix} 0 \\ 2 \end{smallmatrix}\right) \not> 0$, only (ii) of Proposition 7 applies for (BM). In particular, the Guddat method (GuM) becomes

$$\text{minimize } w^T y \text{ subject to } B(y^\circ - y) \geq 0 \text{ and } y \in Y$$

and, again, can be solved as a linear program in the objective space.

It turns out that every closed convex cone $C \in \mathbb{R}^2$ is associated with a linear function and, thus, polyhedral [5]. This implies that both Theorems 1 and 2 apply to all bicriteria programs with respect to a closed convex cone $C \subseteq \mathbb{R}^2$. For multiobjective programs $\mathrm{N}(Y,C)$ with $Y \subseteq \mathbb{R}^m$, $m > 2$, and nonpolyhedral cones $C \subseteq \mathbb{R}^m$, however, only Theorem 2 continues to hold.

7 A Class of Nonpolyhedral Cones

The paper concludes with a discussion of nonpolyhedral cones associated with positively homogeneous functions of the form $\Gamma = (\Gamma_1, \ldots, \Gamma_l) : \mathbb{R}^m \to \mathbb{R}^l$ with $\Gamma_i(c) = A_i^T c - \alpha_i \|B^i c\|$, where $\alpha_i \in \mathbb{R}$, $A_i \in \mathbb{R}^m$, and $B^i \in \mathbb{R}^{\cdot \times m}$, for $i = 1, \ldots, l$. If all $\alpha_i \leq (\geq)\ 0$, then Γ is sublinear (superlinear). In particular, if all $\alpha_i = 0$, then Γ is linear and $C(\Gamma) = C(A)$ with $A = (A_1, \ldots, A_l)^T \in \mathbb{R}^{l \times m}$.

Corollary 1 (to Theorem 2). *Let $\alpha_i \in \mathbb{R}$, $A_i \in \mathbb{R}^m$, and $B^i \in \mathbb{R}^{\cdot \times m}$, for $i = 1, \ldots, l$, and let $C(\Gamma) \subseteq \mathbb{R}^m$ be the nonpolyhedral cone associated with $\Gamma = (\Gamma_1, \ldots, \Gamma_l) : \mathbb{R}^m \to \mathbb{R}^l$ with $\Gamma_i(c) = A_i^T c - \alpha_i \|B^i c\|$. If all $\alpha_i \leq 0$, then*

$$\Gamma[\mathrm{N}(Y, C(\Gamma))] \subseteq \mathrm{N}(\Gamma[Y], \mathbb{R}^l_{\geq}).$$

If all $\alpha_i \geq 0$, the reverse inclusion in Theorem 2 holds if the function Γ is also injective, or if $C(\Gamma)$ is pointed. For special cases, this can be verified easily.

Proposition 9. *Let $\alpha \geq 0$ and $C(\Gamma) \subseteq \mathbb{R}^m$ be the nonpolyhedral cone associated with $\Gamma = (\Gamma_1, \ldots, \Gamma_m) : \mathbb{R}^m \to \mathbb{R}^m$ with $\Gamma_i(c) = c_i - \alpha \|c\|_p$. If $\alpha m^{1/p} < 1$, then the function Γ is injective, and the cone $C(\Gamma)$ is convex and pointed.*

Proof. To show that Γ is injective, let $\Gamma(c) = \Gamma(d)$ and, without loss of generality, assume that $\varepsilon = \alpha(\|c\|_p - \|d\|_p) \geq 0$ (otherwise switch c and d). Then $c_i - \alpha\|c\|_p = d_i - \alpha\|d\|_p$, or $c_i - d_i = \alpha(\|c\|_p - \|d\|_p) = \varepsilon$, and thus, $c = d + \varepsilon e$. Hence, $\|c\|_p = \|d + \varepsilon e\|_p \leq \|d\|_p + \varepsilon\|e\|_p$, or $\varepsilon = \alpha(\|c\|_p - \|d\|_p) \leq \varepsilon\alpha\|e\|_p = \varepsilon\alpha m^{1/p}$, and thus, $\varepsilon = 0$ and $c = d$, showing that Γ is injective. As Γ, in particular, is superlinear, Proposition 3 then gives that $C(\Gamma)$ is convex and pointed. □

Corollary 2 (to Theorem 2, with Proposition 9). *Let $\alpha \geq 0$ and $\alpha m^{1/p} < 1$, and let $C(\Gamma) \subseteq \mathbb{R}^m$ be the nonpolyhedral cone associated with $\Gamma = (\Gamma_1, \ldots, \Gamma_m) : \mathbb{R}^m \to \mathbb{R}^m$ with $\Gamma_i(c) = c_i - \alpha\|c\|_p$. Then*

$$\mathrm{N}(\Gamma[Y], \mathbb{R}^m_{\geq}) \subseteq \Gamma[\mathrm{N}(Y, C(\Gamma))].$$

The formulation of the generating methods (GeM), (GuM), and (BM), and application of Propositions 5, 6 and 7 require knowledge of (at least a subset of) the dual cone $C^+(\Gamma)$, which needs to be derived based on the particularly chosen cone $C(\Gamma)$. This is exemplified in the following proposition.

Proposition 10. *Let $\alpha_i \geq 0$ for all $i = 1,\ldots,m$, and let $C(\Gamma) \subseteq \mathbb{R}^m$ be the convex nonpolyhedral cone associated with $\Gamma = (\Gamma_1,\ldots,\Gamma_m) : \mathbb{R}^m \to \mathbb{R}^m$ with $\Gamma_i(c) = c_i - \alpha_i\|c\|_p$. Then $C(\Gamma)$ belongs to its dual, $C(\Gamma) \subseteq C^+(\Gamma)$, and $e \in \operatorname{int} C^+(\Gamma)$, if some $\alpha_i > 0$.*

Proof. To show that $C(\Gamma)$ belongs to its dual, let $w \in C(\Gamma)$ and choose any $c \in C(\Gamma)$, so $w_i - \alpha_i\|w\|_p \geq 0$ and $c_i - \alpha_i\|c\|_p \geq 0$. Then $w^T c = \sum_{i=1}^m w_i c_i \geq \sum_{i=1}^m \alpha_i^2 \|w\|_p \|c\|_p \geq 0$ and, thus, $w \in C^+(\Gamma)$, showing that $C(\Gamma) \subseteq C^+(\Gamma)$. Furthermore, if $c \neq 0$ and some $\alpha_i > 0$, then $e^T c = \sum_{i=1}^m c_i \geq \sum_{i=1}^m \alpha_i \|c\|_p > 0$ and, thus, $e \in \operatorname{int} C^+(\Gamma)$. □

Hence, for the cone $C(\Gamma)$ in Proposition 10, Propositions 5 and 6 apply for (GeM) and (GuM) with $w \in C(\Gamma) \setminus \{0\}$, and Proposition 7 applies for (BM), if some $\alpha_i > 0$. In particular, the Guddat method (GuM) now becomes

$$\text{minimize } w^T y \text{ subject to } y_i^\circ - y_i - \alpha_i \|y^\circ - y\|_p \geq 0 \text{ for } i = 1,\ldots,m \text{ and } y \in Y$$

and, thus, results in a nonlinear p-th order cone program.

References

1. Benson HP (1978) Existence of efficient solutions for vector maximization problems. J Optim Theory Appl 26(4):569–580
2. Cambini A, Luc DT, Martein L (2003) Order-preserving transformations and applications. J Optim Theory Appl 118(2):275–293
3. Ehrgott M (2005) Multicriteria optimization, 2nd edn. Springer, Berlin
4. Ehrgott M, Wiecek MM (2005) Multiobjective programming. In: Figueira J et al (eds) Multiple criteria decision analysis (Int Ser Oper Res Management Sci 78). Springer, New York, pp 667–722
5. Eichfelder G (2006) Parametergesteuerte Lösung nichtlinearer multikriterieller Optimierungsprobleme. Dissertation, Universität Erlangen-Nürnberg, Germany
6. General Algebraic Modeling System (GAMS) web site: Conic programming in GAMS. Http://www.gams.com/conic/
7. Geoffrion AM (1968) Proper efficiency and the theory of vector maximization. J Math Anal Appl 22(3):618–630
8. Gerth C, Weidner P (1990) Nonconvex separation theorems and some applications in vector optimization. J Optim Theory Appl 67(2):297–320
9. Guddat J, Guerra Vasquez F, Tammer K, Wendler K (1985) Multiobjective and stochastic optimization based on parametric optimization (Math Res 26). Akademie-Verlag, Berlin
10. Hunt BJ, Wiecek MM (2003) Cones to aid decision making in multicriteria programming. In: Tanino T et al (eds) Multiobjective programming and goal programming (Adv Soft Comput). Springer, Berlin, pp 153–158
11. Isac G, Bahya AO (2002) Full nuclear cones associated to a normal cone. Application to Pareto efficiency. Appl Math Lett 15(5):633–639
12. Jahn J (2004) Vector optimization. Theory, applications, and extensions. Springer, Berlin
13. Lobo MS, Vandenberghe L, Boyd S, Lebret H (1998) Applications of second-order cone programming. Linear Algebra Appl 284(1–3):193–228
14. Lotov AV, Bushenkov VA, Kamenev GK (2004) Interactive decision maps. Approximation and visualization of Pareto frontier (Appl Optim 89). Kluwer, Boston

15. Noghin VD (1997) Relative importance of criteria: a quantitative approach. J MultiCrit Dec Anal 6:355–363
16. Pascoletti A, Serafini P (1984) Scalarizing vector optimization problems. J Optim Theory Appl 42(4):499–524
17. Phelps RR (1993) Convex functions, monotone operators and differentiability. Lecture Notes in Math 1364, 2nd edn. Springer, Berlin
18. Rockafellar RT, Wets RJ-B (1998) Variational analysis (Grundlehren Math Wiss [Fundamental Principles of Mathematical Sciences], No. 317). Springer, Berlin
19. Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization (Math Sci Eng No. 176). Academic Press, Orlando, FL
20. Weidner P (2003) Tradeoff directions and dominance sets. In: Tanino T et al (eds) Multiobjective programming and goal programming (Adv Soft Comput). Springer, Berlin, pp 275–280
21. Yu P-L (1974) Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. J Optim Theory Appl 14:319–377
22. Yu P-L (1985) Multiple-criteria decision making. Concepts, techniques, and extensions (Math Concepts Methods Sci Eng No. 30). Plenum, New York

A GP Formulation for Aggregating Preferences with Interval Assessments

Esther Dopazo and Mauricio Ruiz-Tagle

Abstract The purpose of this paper is to develop methods for the estimation of punctual priority weights from interval pairwise comparison matrices given by a group of experts. Because the complexity of real decision problems, the subjective judgments and different patterns of the experts, interval comparison matrices provide a flexible framework to account uncertainty and to achieve consensus solutions among a group of experts. The development of the provided methods relies on l_p-distances to measure the distance between the preference information given by the experts and its normative prototype. Then a minimization problem in the l_p-distance under some constraints is obtained. The proposed approach is made operational with the help of an Interval Goal Programming formulation.

1 Introduction

Let be a group multicriteria decision-making problem, defined by a finite set of alternatives and a group of decision makers (DMs), where preferences are structured by pairwise comparison (pc) matrices.

In many applications, due to incomplete information or knowledge, unquantifiable information, imprecise data, etc., a natural way for expressing preferences is interval assessments. So the problem is to develop methods for the estimation of priority punctual weights from pc matrices with this kind of assessments. In this context, different problems appear simultaneously:

(a) The problem of rationality (i.e., to derive priority weights from pc matrices without consistency properties), introducted by Saaty [15], in the context of the Analytic Hierarchy Process (AHP).

E. Dopazo (✉)
Facultad de Informática, Technical University of Madrid, Campus de Montegancedo, CP28660, Boadilla del Monte (Madrid), Spain
e-mail: edopazo@fi.upm.es

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

(b) The consensus problem (i.e., to integrate preferences from a group of DMs).
(c) The imprecise data management.

There are some studies in the literature on how to deal with interval data in the case of a single expert. In the AHP context some methods, based on eigenvector computation, are introduced in [1, 2, 12, 16, 18]. In [10] a lexicographic goal programming approach is provide for a specific distance. The group problem with interval assessments has been considered in [4] by using a Logarithmic Goal Programming technique. They all provide interval weights. When the interval weights overlap, there is no unique ranking of alternatives. In this case, additional ranking procedures are required in order to compare the final alternatives score. In many applications, a definitive rank order is expected from data given by experts.

On the other hand, a fuzzy approach to the group problem has been considered in [3], [13], and [17]. All of them provide fuzzy priority vectors, and defuzzification techniques are required to produce crisp solutions.

For dealing simultaneously with these problems in a more general pc context, we propose a distance-based framework where a goal programming (GP) formulation will be stated. An analogous model for ranking aggregation has been proposed in [7] and for a group problem with crisp data in [6] and [8].

The paper is organized as follows. Problem formulation in a general distance framework is established in Sect. 2. The Interval Goal Programming formulation of the model is included in Sect. 3. The proposed model is illustrated with the help of a numerical example in Sect. 4. Finally, main conclusions derived from the work are included in Sect. 5.

2 Problem Formulation: A General Distance-Based Approach

Let a finite set of alternatives $X = \{x_1, \ldots, x_n\} (n \geq 2)$ and a group of m experts $\{E_1, \ldots, E_m\}$. We assume that each expert expresses his preferences on the alternatives as an $n \times n$ pc matrix $M^k = (m_{ij}^k)$ [15], where m_{ij}^k represents the estimation of preference ratio between alternatives x_i and x_j given by the expert E_k. The problem is concerned with determining the group priority weights, $w_1, \ldots, w_n$, of the considered alternatives, from preference information given by DMs. We assume that weights are crisp, positive and normalized, i.e.: $\sum_{i=1}^{n} w_i = 1$. We will denote $w = (w_1, \ldots, w_n)^T$ as priority vector.

In the particular case of one expert, because the elements of pc matrix $M = (m_{ij})$ represent estimations of the ratio weights $(\frac{w_i}{w_j})$ of the unknown positive weights, matrix M is expected to satisfy some natural restrictions.

A pc matrix M is said to be reciprocal if $m_{ij} \cdot m_{ji} = 1$ for all $i, j = 1, \ldots, n$ (then automatically $m_{ij} \cdot m_{ji} = 1$). A pc matrix M is said to be consistent if $m_{ij} \cdot m_{jk} = m_{ik}$ for all $i, j, k = 1, \ldots, n$. It is easily seen that consistent matrices are necessarily reciprocal. Saatys theorem [15] states that for every $n \times n$ consistent matrix $M = (m_{ij})$ there exists a set of positive real numbers $w_1, \ldots, w_n$, such that

$$m_{ij} = \frac{w_i}{w_j} \quad i, j = 1, \ldots, n \tag{1}$$

and the vector $w = (w_1, \ldots, w_n)^T$ is unique up to a multiplicative constant. These positive numbers, normalized, define the searched priority weights of the alternatives.

In practice, the complexity of the decision-making problem, imperfect and subjective judgements, etc., lead to non-consistent matrices. This case has been considered in [5] and [11]. Moreover, we consider the group problem: different experts with different interest qualification fields could assign different pairwise data for the same alternatives. In this context, the challenge is to derive group priority weights for the elements of X from possibly non-consistent and not-compatible information given by the experts.

Therefore a theoretical approximation framework is considered to look for a compromise approximated solution. Because the elements of M^k will be considered as perturbations (collecting uncertainty, noise, etc.) of the precise values $\frac{w_i}{w_j}$

$$m_{ij}^k w_j - w_i \approx 0, \quad i, j = 1, \ldots, n; \quad k = 1, \ldots, m, \tag{2}$$

a l_p-distance function will be considered to measure this deviations. In our approach, the aggregation of DMs preferences is addressed to obtain group priority weights that minimize the DMs disagreement in a general l_p-distance framework. Then, we look for the solution that optimizes consensus by minimizing total aggregated deviations from the ideal solution by considering a metric l_p, as follows

$$min \left[\sum_{k=1}^{m} \sum_{i,j=1}^{n} \mid m_{ij}^k w_j - w_i \mid^p \right]^{\frac{1}{p}} \quad if 1 \leq p < \infty \tag{3}$$

where

$$\begin{aligned} &\sum_{i=1}^{n} w_i = 1 \\ &w_i > 0 \qquad \forall i = 1, \ldots, n \end{aligned} \tag{4}$$

For $p = \infty$, we look for the solution that minimizes the maximum deviation

$$\min \max_{\substack{i,j=1,\ldots,n \\ k=1,\ldots,m}} \left(\mid m_{ij}^k w_j - w_i \mid \right) \tag{5}$$

subject to the constraints in (4). The distance parameter p defines the kind of distance metric used and so different decision-making modes. Because p acts on the deviation $\mid m_{ij}^k w_j - w_i \mid$, as p increases more importance is attached to the largest deviation. So, the case $p = 1$ leads to a more robust estimation, whereas the estimation for $p = \infty$ is more sensitive to extreme residual values.

On the other hand, because uncertainty, noise, vagueness of judgments, etc., we consider a more general scenario where imprecise data are allowed. We assume that expert E_k reports his preferences on the elements of X, giving an interval pc matrix on X, $M^k = \left([\underline{m_{ij}^k}, \overline{m_{ij}^k}]\right)$ as follows: he judges that alternative i is between $\underline{m_{ij}^k}$ and $\overline{m_{ij}^k}$ times more important than alternative j with $\underline{m_{ij}^k}, \overline{m_{ij}^k} > 0$ and $\underline{m_{ij}^k} \leq \overline{m_{ij}^k}$. Then, we face to the above optimization problems (3) and (5), in whose data are intervals $m_{ij}^k \in [\underline{m_{ij}^k}, \overline{m_{ij}^k}]$.

Because it is not an easy computational task to solve the above optimizations problems with interval data, in order to make operative the established analytical model, we will consider Interval Goal Programming.

3 A GP Formulation

Once the analytical framework have been established, we focus on computing the approximated weights from the proposed minimization problems (3) and (5) where data given by the experts are interval data. For that purpose, Interval Goal Programming ([7] , [9]) will be considered. We introduce the following deviation variables:

$$\begin{aligned} \underline{m_{ij}^k} w_j - w_i &= \underline{n_{ij}^k} - \underline{p_{ij}^k} \\ \overline{m_{ij}^k} w_j - w_i &= \overline{n_{ij}^k} - \overline{p_{ij}^k} \qquad i,j = 1,\ldots,n, \quad k = 1,\ldots,m \end{aligned} \tag{6}$$

with

$$\begin{aligned} \underline{n_{ij}^k} &= \frac{1}{2}\left[\left|\underline{m_{ij}^k} w_j - w_i\right| + \left(\underline{m_{ij}^k} w_j - w_i\right)\right] \\ \underline{p_{ij}^k} &= \frac{1}{2}\left[\left|\underline{m_{ij}^k} w_j - w_i\right| - \left(\underline{m_{ij}^k} w_j - w_i\right)\right] \\ \overline{n_{ij}^k} &= \frac{1}{2}\left[\left|\overline{m_{ij}^k} w_j - w_i\right| + \left(\overline{m_{ij}^k} w_j - w_i\right)\right] \\ \overline{p_{ij}^k} &= \frac{1}{2}\left[\left|\overline{m_{ij}^k} w_j - w_i\right| - \left(\overline{m_{ij}^k} w_j - w_i\right)\right] \quad i,j = 1,\ldots,n, k = 1,\ldots,m \end{aligned} \tag{7}$$

The variables $\underline{n_{ij}^k}$, $\overline{n_{ij}^k}$, $\underline{p_{ij}^k}$ and $\overline{p_{ij}^k}$ express the negative and the positive deviations, respectively.

In the ideal case, when all estimations are precise and compatible, $\underline{n_{ij}^k} = 0$ and $\overline{p_{ij}^k} = 0, \quad i,j = 1,\ldots,n, \quad k = 1,\ldots,m$. In the general case, non-rational properties of each pc matrix and each decision maker expectations provide incompatible pairwise information. Therefore, following the approach considered in the above section, we look for a group priority vector that minimizes the aggregation of deviations from interval DMs preferences ($M^k, k = 1,\ldots,m$) and the normative prototype $(\frac{w_i}{w_j})$ in a l_p-distance framework. So, we look for approximated positive solutions

that minimize the aggregation of non-wanted residuals, $\underline{n_{ij}^k}$ and $\overline{p_{ij}^k}$, $i,j=1,\ldots,n$, $k=1,\ldots,m$, in a l_p-metric space.

Therefore, for $p \in [1,\infty)$, we obtain the following mathematical programming problem, with the objective function

$$min \left[\sum_{k=1}^{m} \sum_{i,j=1}^{n} \left(\underline{n_{ij}^k} + \overline{p_{ij}^k} \right)^p \right]^{\frac{1}{p}}, \tag{8}$$

subject to:

$$\begin{aligned}
\underline{m_{ij}^k} w_j - w_i - \underline{n_{ij}^k} + \underline{p_{ij}^k} &= 0 \qquad i,j=1,\ldots,n, \quad k=1,\ldots,m \\
\overline{m_{ij}^k} w_j - w_i - \overline{n_{ij}^k} + \overline{p_{ij}^k} &= 0 \qquad i,j=1,\ldots,n, \quad k=1,\ldots,m \\
\sum_{i=1}^{n} w_i &= 1 \\
\underline{n_{ij}^k}, \overline{n_{ij}^k}, \underline{p_{ij}^k}, \overline{p_{ij}^k} &\geq 0 \qquad i,j=1,\ldots,n, \quad k=1,\ldots,m \\
w_i &> 0 \qquad \forall i=1,\ldots,n.
\end{aligned} \tag{9}$$

For $p=\infty$, we obtain the objective function

$$minD$$

subject to $\underline{n_{ij}^k} + \overline{p_{ij}^k} \leq D$, $i,j=1,\ldots,n$, $k=1,\ldots,m$ and goals and constraints of the model (9). $\overline{D}$ is an extra positive variable that quantifies the maximum deviation, it indicates maximum disagreement with respect to the consensus achieved.

We should make some important points from a computational point of view, for the most used values of p. For $p=1$ and $p=\infty$, the above formulations are reduced to linear programming problems that can be solved by using the simplex method. The case $p=2$ corresponds to a quadratic programming problem for which several numerical tools are available.

4 Numerical Example

We present a numerical example to illustrate the proposed method. We suppose we are in a fictitious early stage of a national forest planning process, where the main objective is to estimate future wood supply from commercial planting forests in Chile. A set of four alternatives are presented to a group of four experts in forestry economics, from several organizations (private, academic and research) in Chile. They assess their preferences about the alternatives by the pc interval matrices M^1, M^2, M^3 and M^4, using the Saaty's scale, as follows:

Table 1 Priority vectors and their associated rankings for $p = 1, 2, \infty$

	Metric		
	$p=1$	$p=2$	$p=\infty$
w	$\begin{pmatrix} 0.417 \\ 0.083 \\ 0.417 \\ 0.083 \end{pmatrix}$	$\begin{pmatrix} 0.500 \\ 0.119 \\ 0.285 \\ 0.096 \end{pmatrix}$	$\begin{pmatrix} 0.446 \\ 0.161 \\ 0.268 \\ 0.125 \end{pmatrix}$
Ranking	$x_1 = x_3 \succ x_2 = x_4$	$x_1 \succ x_3 \succ x_2 \succ x_4$	$x_1 \succ x_3 \succ x_2 \succ x_4$

$$\mathbf{M^1} = \begin{pmatrix} [1,1] & [5,9] & \left[\frac{1}{5},\frac{1}{3}\right] & [3,5] \\ \left[\frac{1}{9},\frac{1}{5}\right] & [1,1] & \left[\frac{1}{9},\frac{1}{5}\right] & \left[\frac{1}{7},\frac{1}{5}\right] \\ [3,5] & [5,9] & [1,1] & [3,7] \\ \left[\frac{1}{5},\frac{1}{3}\right] & [5,7] & \left[\frac{1}{7},\frac{1}{3}\right] & [1,1] \end{pmatrix} \quad \mathbf{M^2} = \begin{pmatrix} [1,1] & [1,3] & [3,5] & [3,5] \\ \left[\frac{1}{3},1\right] & [1,1] & \left[\frac{1}{7},\frac{1}{3}\right] & \left[\frac{1}{5},\frac{1}{3}\right] \\ \left[\frac{1}{5},\frac{1}{3}\right] & [3,7] & [1,1] & [1,5] \\ \left[\frac{1}{5},\frac{1}{3}\right] & [3,5] & \left[\frac{1}{5},1\right] & [1,1] \end{pmatrix} \tag{10}$$

$$\mathbf{M^3} = \begin{pmatrix} [1,1] & [3,5] & \left[\frac{1}{5},\frac{1}{3}\right] & [3,5] \\ \left[\frac{1}{5},\frac{1}{3}\right] & [1,1] & \left[\frac{1}{5},\frac{1}{3}\right] & \left[\frac{1}{7},\frac{1}{3}\right] \\ [3,5] & [3,5] & [1,1] & [5,7] \\ \left[\frac{1}{5},\frac{1}{3}\right] & [3,7] & \left[\frac{1}{7},\frac{1}{5}\right] & [1,1] \end{pmatrix} \quad \mathbf{M^4} = \begin{pmatrix} [1,1] & [3,5] & [1,3] & [5,7] \\ \left[\frac{1}{5},\frac{1}{3}\right] & [1,1] & \left[\frac{1}{5},\frac{1}{3}\right] & \left[\frac{1}{5},\frac{1}{3}\right] \\ \left[\frac{1}{3},1\right] & [3,5] & [1,1] & [3,5] \\ \left[\frac{1}{7},\frac{1}{5}\right] & [3,5] & \left[\frac{1}{5},\frac{1}{3}\right] & [1,1] \end{pmatrix}$$

In this example, we find that matrices given by the experts are reciprocal, but information provided by them is discrepant and not compatible. For instance, the range of estimations about options x_1 and x_3 given by the experts E_1 and E_2 are different. Even the intersection is empty, $\left[\underline{m^1_{13}}, \overline{m^1_{13}}\right] \cap \left[\underline{m^2_{13}}, \overline{m^2_{13}}\right] = \emptyset$. So, a consensus between their preference information is required.

We applied the proposed method using l_p-metrics for the most usual values of p. We obtained the priority vectors and associated rankings listed in Table 1.

We notice that there are ties between options x_1 and x_3, and x_2 and x_4 for metric 1. These ties are solved in the result obtained with $p = 2$ and $p = \infty$. They yield dominance for option x_1 over x_3 and for x_2 over x_4, because greater deviations between each pairwise comparison are emphasized.

We developed a GP matrix generator using MS Visual FoxPro and problems were optimized using ILOG CPLEX (Java classes).

5 Conclusions

Our objective is to provide some methods for the estimation of priority punctual weights that best reflects imprecise preference opinions of a group of DMs in an interval pc scenario.

Because rational properties and unanimous consensus are not guaranteed in real-life problems, a general approximation framework is considered to define a compromise solution. The pairwise data given by the experts are considered as samples of an ideal consistent ratio-matrix. For that purpose, l_p-metrics are considered to measure the distance from the experts data to the normative ones. This general framework defines different decision-making modes. Moreover, in this context the p-parameter can be interpreted as having a preference and consensus meaning.

The proposed approach allows dealing with interval assessments which is a natural way for an expert to express his views in presence of imprecise and unquantifiable information. Besides, it provides flexibility that can be exploited to achieve a compromise solution between the experts in a group problem.

The given Interval GP formulation provides flexible and efficient methods that can be easily implemented for computing priority crisp weights for the most usual values of p in practical applications.

On the other hand, most of methods dealing with interval data in the literature, lead to interval weights. When the interval weights overlap, there is not a unique ranking of alternatives. Then, additional ranking procedures are required in order to compare the final alternatives score.

Acknowledgements The authors would like to thank Prof. Jacinto Gonzlez-Pachn for his helpful comments. Thanks are also given to the anonymous reviewers for their suggestions.

The research is partly supported by Project MTM2007-67232, "Ministerio de Educacin y Ciencia" of Spain.

References

1. Arbel A (1989) Approximate articulation of preference and priority derivation. Eur J Oper Res 43(3):317–326
2. Arbel A, Vargas L (1993) Preference simulation and preference programming: robustness issues in priority derivation. Eur J Oper Res 69(2):200–209
3. Boender CGE, de Graan JG, Lootsma FA (1989) Multicriteria decision analysis with fuzzy pairwise comparisons. Fuzzy Sets Syst 29(2):133–143
4. Bryson NKM, Joseph A (2000) Generating consensus priority interval vectors for group decision-making in the AHP. J MultiCrit Decision Anal 9(4):127–137
5. Dopazo E, González-Pachón J (2003) Consistency-driven approximation of a pairwise comparison matrix. Kybernetyca 39(5):561–568
6. Dopazo E, González-Pachón J, Robles J (2005) A distance-based method for preference information retrieval in paired comparisons. Lect Notes Comput Sci 3646:66–73
7. González-Pachón J, Romero C (2001) Aggregation of partial ordinal rankings: an interval goal programming approach. Comput Oper Res 28(8):827–834
8. González-Pachón J, Romero C (2007) Inferring consensus weights from pairwise comparison matrices with suitable properties. Ann Oper Res 154:123–132
9. Ignizio JP (1982) Linear programming in single- and multiple-objective systems. Prentice-Hall, Englewood Cliffs
10. Islam R, Biswal MP, Alam SS (1997) Preference programming and inconsistent interval judgments. Eur J Oper Res 97(1):53–62
11. Jones DF, Mardle SJ (2004) A distance-metric methodology for the derivation of weights from a pairwise comparison matrix. J Oper Res Soc 55:869–875

12. Kress M (1991) Approximate articulation of preference and priority derivation - a comment. Eur J Oper Res 52:382–383
13. Mikhailov L (2004) Group prioritization in the AHP by fuzzy preference programming method. Comput Oper Res 31(2):293–301
14. Romero C (1991) Handbook of critical issues in goal programming. Pergamon, Oxford
15. Saaty TL (1980) The analytic hierarchy process. Mc-Graw Hill, New York
16. Salo AA, Hämäläinen RP (1995) Preference programming through approximate ratio comparisons. Eur J Oper Res 82:458–475
17. Yu C-S (2002) A GP-AHP method for solving group decision-making fuzzy AHP problems. Comput Oper Res 29:1969–2001
18. Wang Y-M, Yang J-B, Xu D-L (2005) Interval weight generation approaches based on consistency test and interval comparison matrices. Appl Math Comput 167(1):252–273

Bicriterion Shortest Paths in Stochastic Time-Dependent Networks

Lars Relund Nielsen, Daniele Pretolani, and Kim Allan Andersen

Abstract In recent years there has been a growing interest in using stochastic time-dependent (STD) networks as a modelling tool for a number of applications within such areas as transportation and telecommunications. It is known that an optimal routing policy does not necessarily correspond to a path, but rather to a time-adaptive strategy. In some applications, however, it makes good sense to require that the routing policy should correspond to a loopless path in the network, that is, the time-adaptive aspect disappears and a priori route choice is considered.

In this paper we consider bicriterion a priori route choice in STD networks, i.e. the problem of finding the set of efficient paths. Both expectation and min–max criteria are considered and a solution method based on the two-phase method is devised. Experimental results reveal that the full set of efficient solutions can be determined on rather large test instances, which is in contrast to the time-adaptive case.

Keywords: Stochastic time-dependent networks · Bicriterion shortest path · Two-phase method · Computational analysis

L.R. Nielsen (✉)
Department of Genetics and Biotechnology, University of Aarhus, P.O. Box 50, DK-8830 Tjele, Denmark
e-mail: lars@relund.dk

D. Pretolani,
Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, Via Amendola 2, I-42100 Reggio Emilia, Italy,
e-mail: daniele.pretolani@unimore.it

K.A. Andersen,
Department of Business Studies, University of Aarhus, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark
e-mail: kia@asb.dk

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

1 Introduction

Recently there has been a growing focus on stochastic time-dependent networks* (*STD networks*) which often provide a better modelling tool in e.g. transportation applications [3, 9, 10, 12]. Here travel times are modeled as random variables with time-dependent distributions. Particular cases, such as non-stochastic time-dependent networks (see [5] for a recent overview) and time-independent stochastic networks (see [15]) have been widely studied, but are not considered here.

STD networks were first addressed by Hall [4] who pointed out several ways to formulate the route selection problem in STD networks. If the driver is allowed to react to revealed (actual) arrival times at intermediate nodes, the best route is not necessarily a path, but rather a time-adaptive *strategy* that assigns optimal successor arcs to a node as a function of leaving time. This is referred to as *time-adaptive route choice*. If a *loopless path* must be specified before travel begins, and no deviations from the route are permitted, the path is selected *a priori* on the basis of only the probability distributions of the arc travel-times. This is referred to as *a priori route choice* and may be useful for routing highly sensitive substances for which the path travelled must be preapproved. The problem of finding a minimum expected travel time path under a priori route choice is NP-hard [14].

It is quite obvious that multicriteria a priori route choice in STD networks is a relevant and difficult problem. However, only two papers exist on the subject. Miller-Hooks and Mahmassani [6] consider bicriterion a priori route choice in discrete STD networks, the objectives being minimizing expected travel time and cost. A label-correcting procedure is described, which guarantees that all the efficient paths can be obtained. Computational results are presented on a single road network. Chang et al. [1] consider multicriterion a priori route choice in a continuous time STD network, where travel times are normally distributed. They devise a heuristic method based on the first two moments of the distributions, where an approximate stochastic dominance criterion is adopted to compare paths. Computational results are presented on an example network and a single road network.

In this paper we consider bicriterion route choice problems in STD networks under a priori route choice. More specifically we consider the problem of finding the set of efficient paths between an origin and a destination node, when leaving the origin at time zero. We assume that departure times are integer and that travel times are discrete random variables. The paper differs from previous work in the following aspects:

(a) We propose a new algorithm using the two-phase method to determine the set of efficient paths as opposed to the labelling approach proposed by Miller-Hooks and Mahmassani [6].
(b) We perform a reasonably wide computational experience on grid networks, where we address the case of two cost criteria, besides the (somehow easier)

* Also known as random time-dependent networks, stochastic time-varying networks or stochastic dynamic networks.

time/cost case; this allows us to evaluate the effect of uncorrelated and correlated cost criteria; furthermore, we consider both expectation and min–max criteria.

(c) In our computational setting we do not consider a "steady state" with deterministic travel times at the end of a single peak period in contrast to [6]. During the travel, times are stochastic and time dependent, and several peak periods are encountered.

(d) Since our algorithms solve the bicriterion problem exactly on the set of instances addressed here, we are able to compare the nondominated points found under a priori and time-adaptive route choice.

The paper is organized as follows. In Sect. 2 we briefly introduce STD networks. In Sect. 3 we give a short description of the two-phase method, and describe the procedure we use for its implementation. Computational results are reported in Sect. 4.

2 Stochastic Time-Dependent Networks

Let $G = (N,A)$ be a directed graph, referred to as *the topological network* and let o, $d \in N$ denote two different nodes which represent the *origin* and the *destination* node in G, respectively. Throughout this paper we consider routing from o towards d when leaving node o at time zero.

Departure and arrival times belong to a finite *time horizon*, i.e. a set $H = \{0,1,...,t_{\max}\}$. Let $X(u,v,t)$ denote the arrival time at node v when leaving node u at time t along arc (u,v) which is a discrete random variable with density

$$\Pr\left(X\left(u,v,t\right) = t_i\right) = \theta_{uvt}\left(t_i\right), \quad t_i \in I\left(u,v,t\right)$$

where $I(u,v,t) = \{t_1,...,t_{\kappa(u,v,t)}\}$ denotes the set of $\kappa(u,v,t)$ possible arrival times at node v. We assume that travel times are positive, and no waiting is allowed in the nodes. For the situation where waiting is allowed see [11].

Costs are considered in the model by letting $c(u,v,t)$ denote the *travel cost* of leaving node u at time t along arc (u,v) and $g(t)$ the *penalty cost* of arriving at node d at time t.

A *strategy* S is a function which provides routing choices for travelling from o at time zero towards d. That is, if $S(u,t) = (u,v)$ a traveller leaving node u at time t travels along arc (u,v). Note that a strategy must provide routing choice for all possible arrival times at an intermediate node. Under time-adaptive route choice finding the best route with respect to some criterion corresponds to finding the best strategy. Under a priori route choice we must travel along a loopless path in G, that is, we must adopt a *path-strategy*, where the successor arcs of a node for different leaving times are time-independent. From now on, we shall identify paths with path-strategies. Let us denote by $\mathscr{S}$ the set of all strategies and with $\mathscr{S}_P$ the set of all path-strategies. Clearly, $\mathscr{S}_P \subseteq \mathscr{S}$.

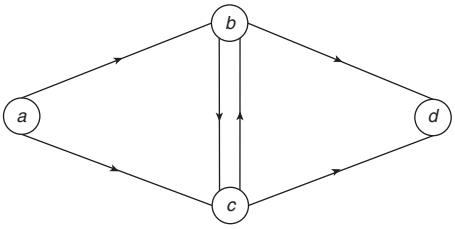


Fig. 1 The topological network G

Table 1 Input parameters

$(u,v),t$	$(a,b),0$	$(a,c),0$	$(b,c),1$	$(b,c),3$	$(b,d),1$
$I(u,v,t)$	$\{1,3\}$	$\{2\}$	$\{2\}$	$\{4,5\}$	$\{3\}$
$c(u,v,t)$	$(1,1)$	$(3,0)$	$(2,2)$	$(0,4)$	$(3,8)$
$(u,v),t$	$(b,d),3$	$(c,b),2$	$(c,d),2$	$(c,d),4$	$(c,d),5$
$I(u,v,t)$	$\{6,7\}$	$\{3\}$	$\{3,4\}$	$\{5,6\}$	$\{6\}$
$c(u,v,t)$	$(3,6)$	$(2,1)$	$(4,2)$	$(3,3)$	$(1,5)$

Example 1. Consider the topological network in Fig. 1, where a is the origin node and d is the destination node. For each arc in G, the possible departure and arrival times are listed in Table 1. Here a pair $((u,v),t)$ corresponds to a possible leaving time t from node u along arc (u,v). For the sake of simplicity, we assume that $X(u,v,t)$ has a uniform density, i.e., for each $t' \in I(u,v,t)$, we have $\theta_{uvt}(t') = 1/|I(i,j,t)|$. For example, if we leave node b at time 3 along arc (b,c), we arrive at node c at time 4 or 5 with the same probability $1/2$. Two possible strategies are

$$\begin{aligned} S_1 &: S_1(a,0) = (a,b),\ S_1(b,1) = (b,d),\ S_1(b,3) = (b,d); \\ S_2 &: S_2(a,0) = (a,b),\ S_2(b,1) = (b,d),\ S_2(b,3) = (b,c), \\ &\quad S_2(c,4) = (c,d),\ S_2(c,5) = (c,d). \end{aligned}$$

Strategy S_1 is a path-strategy and corresponds to the path $a-b-d$ while for strategy S_2 we travel different routes depending on the leaving time from node b.

In this paper we assume that two values are associated with a strategy, namely travel time and cost, where cost is considered in general terms, e.g., a risk measure or the economic travel cost. Furthermore, different criteria are considered, namely, *expectation criteria* (minimize expected travel time or cost) and *min–max criteria* (minimize maximum possible travel time or cost). The value associated with a strategy according to these criteria can be formally defined by means of sets of recursive equations, see [14]. Given a strategy $S \in \mathscr{S}_P$ let $W(S) = (W_1(S), W_2(S))$ denote the two-dimensional vector, where $W_i(S)$, $i = 1,2$ is the value associated with S with respect to one of the above four criteria. In this paper we face the following problem:

$$\begin{aligned} &\min W(S) = (W_1(S), W_2(S)) \\ &\text{s.t.} \quad S \in \mathscr{S}_P. \end{aligned} \tag{1}$$

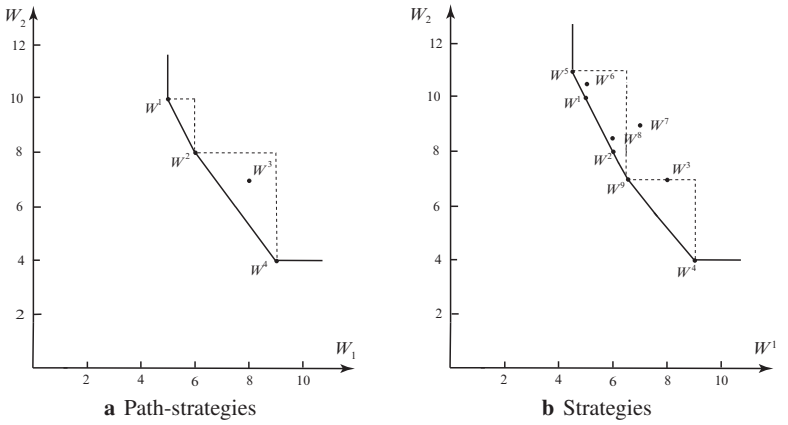
That is, we want to find the set of *efficient* (Pareto optimal) path-strategies $\mathscr{S}_E \subseteq \mathscr{S}_P$ or equivalently the set of *nondominated* points $\mathscr{W}_E = \{W(S) \in \mathbb{R}^2 \mid S \in \mathscr{S}_E\}$ in the criterion space $\mathscr{W}$. We will follow the predominant thought within bicriterion optimization which is to identify all nondominated points, providing one corresponding efficient solution for each nondominated point.

Nondominated points can be partitioned into two sets, namely supported and unsupported. The supported ones can be further subdivided into extreme and nonextreme as illustrated in the following example.

Example 1. (continued) Assume that two costs $c_i(u,v,t)$, $i = 1,2$, are given for each leaving time t from node u along arc (u,v), see Table 1.

Consider (1) when both criteria are to minimize expected cost. The criterion points corresponding to the four possible loopless paths in G are illustrated in Fig. 2a. In this example all four points are nondominated. W^1, W^2 and W^4 are supported points all of which are extreme. The extreme points define two triangles, shown with dashed lines, in which it may be possible to find unsupported nondominated points such as W^3.

In general the total number of path-strategies is significantly lower than the number of strategies. Under time-adaptive route choice, i.e. we consider (1) with the constraint replaced with $S \in \mathscr{S}$ instead, we have nine possible strategies, i.e. five points do not correspond to a path. All nine points are illustrated in Fig. 2b. Under time-adaptive route choice the five points W^1, W^2, W^4, W^5 and W^9 are supported nondominated points of which W^1 and W^2 are non-extreme. Points which do not lie inside the triangles such as W^3 and W^7 are dominated. Moreover, the two points W^6



a Path-strategies **b** Strategies

Fig. 2 Criterion spaces under a priori and time-adaptive route choice

and W^8 are dominated by W^1 and W^2, respectively. Further details and illustrations of the (path-)strategies can be found in [11].

Please note that the set of nondominated points under time-adaptive route choice will always dominate the set of nondominated points under a priori route choice. However, a nondominated point under a priori route choice (such as W^3) may be dominated, if time-adaptive route choice is considered. ■

3 Solution Method

In this section we devise a solution method finding all efficient paths under a priori route choice based on the two-phase method. The two-phase approach is a general method for solving bicriterion discrete optimization problems such as (1). As the name suggests, the two-phase method divides the search for nondominated points into two phases. In phase one, extreme supported nondominated points are found. These extreme points define a number of triangles in which unsupported nondominated points may be found in phase two. For a description of a generic two-phase method see [13].

Both phases make use of a parametric function $\gamma\colon (\mathscr{W}, \mathbb{R}_+) \to \mathbb{R}_+$ which denotes the *parametric cost* of a path-strategy $S \in \mathscr{S}_{\mathscr{P}}$.

$$\gamma(W(S), \lambda) = W_1(S)\lambda + W_2(S). \tag{2}$$

It is well-known that given $\lambda > 0$ the path-strategy S with minimum parametric cost $\gamma(W(S), \lambda)$ corresponds to a supported nondominated point and hence is efficient. As a result all supported extreme nondominated points can be found in phase one by solving (2) for different values of λ, see [2].

Phase two searches each triangle using an algorithm for ranking path-strategies with respect to the parametric weight (2), where λ is a function of the slope of the line joining the two points defining the triangle. The search stops when the parametric weight reaches an upper bound, which in turn is dynamically updated (decreased) when new nondominated points are found.

It must be kept in mind that in both phases we have to solve a sequence of difficult problems, since a priori routing even for the single criterion case is NP-hard. In order to solve these problems we adopted the newly developed algorithm for ranking paths in STD networks (*procedure* K-BPS), see [10]. However, the effectiveness of this approach is quite different for expectation and min–max criteria.

As long as two expectation criteria are used, the parametric cost $\gamma(W(S), \lambda)$ of a strategy S is equal to the cost of S with respect to the cost vector $c_\lambda = c_1 \cdot \lambda + c_2$. This result has been proved in [9, Th. 2] for strategies, and clearly holds for path-strategies too. As a consequence, we can rank paths with respect to the parametric cost by applying procedure *K-BPS* with the costs c_λ. This procedure is also used in phase one, stopping as soon as the best parametric path is found.

Unfortunately, the above result does not hold, if min–max criteria are considered. In this case, c_λ only provides us with a *lower bound* $W_\lambda(S) \leq \gamma(W(S), \lambda)$ (see [9, Th. 3, 4]). By applying procedure *K-BPS* with costs c_λ we generate path-strategies in non-decreasing order of $W_\lambda(S)$. In phase one, for each vale value of λ we let procedure *K-BPS* run until $W_\lambda(S)$ reaches the parametric cost of the best path-strategy generated so far. In phase two a triangle is searched until the lower bound $W_\lambda(S)$ reaches the upper bound of the triangle. In phase one it may happen, due to ranking according to $W_\lambda(S)$, that procedure *K-BPS* generates many paths that actually fall inside the triangle defined by a certain λ. In order to take advantage of this fact, in our computational tests, we adopt a *hybrid* algorithm, where the two phases are combined. More precisely, when a new triangle is identified in the first phase we search inside the triangle by letting procedure *K-BPS* continue until the lower bound $W_\lambda(S)$ reaches the upper bound.

4 Computational Experience

We implemented the algorithm in C++ and compiled the source code with the GNU C++ compiler with optimize option -O on a 1 GHz PIII computer with 1 GB RAM using a Linux Red Hat operating system. The main goals and results of our computational experience have been anticipated in the introduction, where we pointed out the original contributions of our work.

4.1 Test Instances

The STD network test instances are generated using the newly developed generator TEGP (*Time-Expanded Generator with Peaks*) which includes several features inspired by typical aspects of road networks. For more details see [8].

Two grid graphs, with sizes 5×8 and 10×10, are considered; the length of the time horizon is 144 and 288, respectively. Each grid arc is randomly assigned an *off peak* mean travel time, so that the mean travel time changes as a function of leaving time, increasing up to 100% during *peak periods*. For the 5×8 grid the mean travel time follows the two-peaks pattern shown by the dotted line in Fig. 3, while

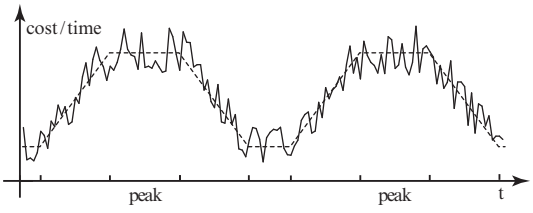


Fig. 3 Peak effect and random perturbation for an arc

for the 10×10 grid the pattern in Fig. 3 is repeated, obtaining four peaks. The travel time distribution is randomly generated around the travel time mean, as a discrete approximation of a normal distribution. Arc costs are generated independently from travel times and increase due to the peak effect with a further *random perturbation* (10% at most) thus following a pattern similar to the solid line in Fig. 3.

The network parameters are summarized in Table 2. Here I_T and I_C denote the range of possible travel times and costs, and κ is the average size of the travel time distributions. For further details about the test instances see [11].

We report results on three different combinations, namely: *T/C*, corresponding to travel time and cost; *C/C* neg_{cor}, where the two costs are negatively correlated; and C/C no_{cor}, where no correlation between the two costs is assumed. Recall that *C/C* neg_{cor} is usually considered to be harder. In all combinations the penalty costs are zero. For each setting of costs five test instances where generated.

4.2 Results

We first consider the case of two expectation criteria. The results are reported in Table 3. Here $|\mathscr{W}_{SE}|$ is the number of supported extreme nondominated points and CPU_{SE} is the average CPU time (in seconds) used to find the supported nondominated points. $|\triangle|$ is the number of triangles, while $|\mathscr{W}_{\triangle}|$ is the number of nondominated points in a triangle (points defining the triangle not included). Finally, $\text{CPU}_{\triangle}$ is the CPU time used to find the nondominated points in a triangle (average and maximum).

In phase one all extreme supported nondominated points can be determined in a reasonable amount of time. The same holds true for phase two (which is the most

Table 2 Test parameters

Size	H	Peaks	I_T	I_C	κ
5×8	144	2	[3,20]	[1,2200]	6
10×10	288	4	[3,20]	[1,2200]	6

Table 3 Results expectation criteria

		$\lvert\mathscr{W}_{SE}\rvert$	CPU_{SE}	$\triangle$	$\lvert\mathscr{W}_{\triangle}\rvert$		$\text{CPU}_{\triangle}$	
					ave	max	ave	max
5×8	T/C	5	0.54	4	1	4	0.16	0.33
5×8	C/C no_{cor}	4	0.57	3	1	4	0.20	0.40
5×8	C/C neg_{cor}	8	1.45	7	3	11	0.73	2.22
10×10	T/C	6	8.90	5	2	8	3.79	22.91
10×10	C/C no_{cor}	8	21.45	7	3	17	5.76	47.57
10×10	C/C neg_{cor}	11	25.04	10	6	26	12.08	43.05

time-consuming phase). That is, we can find the nondominated set for all the test instances considered.

Comparing the different combinations of criteria, we see that the time–cost case is in general easier than the cost–cost cases, and that negatively correlated costs are harder than uncorrelated costs. The plots in Fig. 4 give an intuition of the difference between *C/C* neg_{cor} and $C/C\ no_{cor}$. Indeed, negatively correlated costs produce more nondominated points that are spread in a larger area of the criterion space. As a result, we have more triangles to search, and it takes longer time to search each one of them; see [11] for further details. This fact was also observed under time-adaptive route choice [9] and is a general feature for discrete bicriterion optimization problems, see e.g. [13].

The results for two min–max criteria are presented in Table 4; we only consider the (more difficult) cost/cost combinations here. Compared to expectation criteria the total number of nondominated points is about the same in average. However, the CPU time spent is considerably higher, as we may expect, since the parametric problem is harder to solve and the lower bound used for ranking is not very tight. Also in this case, the problems with negatively correlated costs are much more difficult.

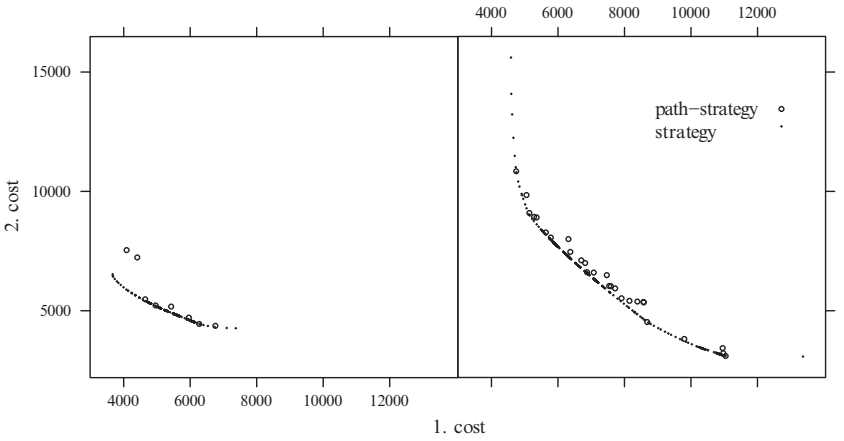


Fig. 4 $\mathscr{W}_E$ for an uncorrelated (left) and negatively correlated (right) test instance (both criteria are minimizing expected cost)

Table 4 Results min–max criteria

		$\lvert\mathscr{W}_{SE}\rvert$	$\triangle$	$\lvert\mathscr{W}_{\triangle}\rvert$		$\text{CPU}_{\triangle}$	
				ave	max	ave	max
5×8	C/C no_{cor}	4	3	1	4	0.32	0.68
5×8	C/C neg_{cor}	7	6	4	14	2.01	5.65
10×10	C/C no_{cor}	8	7	2	12	17.91	80.12
10×10	C/C neg_{cor}	10	9	7	59	100.42	439.72

4.3 Comparison to the Time-Adaptive Case

Comparing the results for the a-priori case to previous results for the time-adaptive case [7, 9] allows us to point out interesting differences. Here we restrict ourselves to expectation criteria, since for min–max criteria the approximation found in [9] is usually rather week.

First of all, recall that the set of nondominated points has been found for all the instances considered. This is in deep contrast to time-adaptive route choice, where not even an ε-approximation with $\varepsilon = 1\%$ could be found for the same set of instances [9]. This result may be viewed as surprising, since finding the best strategy in the single criterion case is easy (can be done in linear time) while finding the best path-strategy is NP-hard. A reasonable explanation of this apparent paradox is that the solution space is much more dense in the time-adaptive case, that is, the total number of path-strategies is much lower than the total number of strategies. Therefore the ranking procedure used in the second phase does not have to rank as many solutions.

In order to get a deeper insight in this issue, we made plots comparing the nondominated set for the a priori case with an approximation of the nondominated set for the time adaptive case, obtained using the algorithms from [9]. Fig. 4 shows two instances on a 5×8 grid with uncorrelated costs (left) and negatively correlated costs (right).

First, as noted above, negatively correlated costs produce more nondominated points, spread in a wider area; this situation arises for both a priori and adaptive routing.

Second, in some cases the a priori nondominated set may contain points close to the time-adaptive nondominated set. Hence solutions found when priori routing must be adopted, due e.g. to outside regulations, may still be as good as those found without this regulation. However, in other cases you may have to pay higher costs, if paths must be adopted (e.g. if the first cost is below 4,500 in the left plot in Fig. 4).

Finally, in general for our instances there are large variations in the values of ε for which the a priori nondominated set turns out to ε-dominate (see [16]) the time-adaptive nondominated set. On average $\varepsilon = 0.1$, the minimum ε value found was 0.03 and the maximum 0.25.

References

1. Chang TS, Nozick LK, Turnquist MA (2005) Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments. Transport Sci 39(3):383–399
2. Cohen J (1978) Multiobjective programming and planning. Academic Press, New York
3. Gao S, Chabini I (2006) Optimal routing policy problems in stochastic time-dependent networks. Transport Res B 40:93–122
4. Hall RW (1986) The fastest path through a network with random time-dependent travel times. Transport Sci 20(3):182–188

5. Hamacher HW, Ruzika S, Tjandra SA (2006) Algorithms for time-dependent bicriteria shortest path problems. Discrete Optim 3:238–254
6. Miller-Hooks E, Mahmassani HS (1998) Optimal routing of hazardous materials in stochastic, time-varying transportation networks. Transport Res Rec 1645:143–151
7. Nielsen LR (2004) Route choice in stochastic time-dependent networks. PhD Dissertation, Department of Operations Research, University of Aarhus URL http://www.imf.au.dk/publs?id=499
8. Nielsen LR (2006) Tegp - time-expanded generator with peaks, February 2006. URL http://www.research.relund.dk. Reference manual
9. Nielsen LR, Andersen KA, Pretolani D (2003) Bicriterion shortest hyperpaths in random time-dependent networks. IMA J Manage Math 14(3):271–303
10. Nielsen LR, Pretolani D, Andersen KA (2004) *K* Shortest paths in stochastic time-dependent networks. Working Paper WP-L-2004-05, Department of Accounting, Finance and Logistics, Aarhus School of Business, Denmark
11. Nielsen LR, Pretolani D, Andersen KA (2006) Bicriterion a priori route choice in stochastic time-dependent networks. Working Paper WP-L-2006-10, Department of Business Studies, Aarhus School of Business, Denmark
12. Opasanon S, Miller-Hooks E (2006) Multicriteria adaptive paths in stochastic time-varying networks. Eur J Oper Res 173:72–91
13. Pedersen CR, Nielsen LR, Andersen KA. The Bicriterion Multi Modal Assignment Problem: Introduction, Analysis, and Experimental Results. Informs Journal on Computing 2008, 20, page 400–411.
14. Pretolani D (2000) A directed hypergraph model for random time-dependent shortest paths. Eur J Oper Res 123(2):315–324
15. Provan JS (2003) A polynomial-time algorithm to find shortest paths with recourse. Networks 41(2):115–125
16. Warburton A (1987) Approximation of pareto optima in multiple-objective, shortest-path problems. Oper Res 35(1):70–79

Clusters of Non-dominated Solutions in Multiobjective Combinatorial Optimization: An Experimental Analysis

Luís Paquete and Thomas Stützle

Abstract This paper presents an analysis on the level of clustering of approximate non-dominated solutions for several instances of the Biobjective Travelling Salesman Problem and the Biobjective Quadratic Assignment Problem. The sets of approximate non-dominated solutions are identified by high performing stochastic local search algorithms. A cluster is here defined as a set of non-dominated solutions such that for any solution of the set there is at least one other that is at a maximum distance k for a given neighborhood function. Of particular interest is to find k for which all approximate non-dominated solutions are in a single cluster. The results obtained from this analysis indicate that the degree of clustering depends strongly on the problem but also on the type of instances of each problem. These insights also suggest that different general-purpose search strategies should be used for the two problems and also for different instance features.

Keywords: Combinatorial optimization · Local search · Multiobjective programming

1 Introduction

When applied to multiobjective combinatorial optimization problems defined in terms of Pareto optimality, local search algorithms return a set of non-dominated solutions that approximate the Pareto optimal set. Several of these algorithms consist of two main steps [1, 6, 9, 12]. In the first step, optimal or approximate solutions with respect to several scalarizations of the objective function vector are obtained;

L. Paquete
University of Coimbra, Department of Informatics Engineering, CISUC, Polo II, 3030-290 Coimbra, Portugal
e-mail: paquete@dei.uc.pt

T. Stützle (✉)
Université Libre de Bruxelles, CoDE, IRIDIA, CP194/6, Avenue Franklin Roosevelt 50, 1050 Brussels, Belgium
e-mail: stuetzle@ulb.ac.be

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

then, in the second step, non-dominated solutions are searched with respect to some neighborhood. Obviously, the success of such algorithms strongly depends on how *close* are high quality solutions in the solution space with respect to the neighborhood chosen.

In this paper, we report and interpret some statistics collected on *clusters* of approximate non-dominated solutions for two multiobjective combinatorial optimization problems in order to provide more insight into potentially successful search strategies for those problems. Informally, a cluster is defined here as a set of solutions that are reachable from each other by applying k elementary moves with respect to a given neighborhood. We then compute the percentage of solutions in the clusters, as well as the number of clusters, for different k. This analysis is performed on several instances of the biobjective traveling salesman problem (BTSP) and the biobjective quadratic assignment problem (BQAP) for various sizes and different features. The set of solutions examined here are taken from the union of a large set of solutions returned by many runs of several high-performing stochastic local search algorithms for these problems described in [11].

2 Clusters of Non-Dominated Solutions

For this analysis, we extend the notion of *efficient graph* given in [5] by introducing a distance δ that corresponds to the minimum number of solutions that must be visited in order to reach a solution u from a solution v, according to a given neighborhood structure $\mathcal{N}$. Given a set of non-dominated solutions for some problem instance and a distance function δ, let $\mathcal{G}_k = (V,E)$ be a graph where V corresponds to the set of non-dominated solutions and an edge in E connects u and v if $\delta(u,v) \leq k$, $u,v \in V$. Then, a cluster of non-dominated solutions at distance k is a *connected component of* $\mathcal{G}_k$. Intuitively, given a single solution in a cluster, a local search may reach any of the solutions inside the cluster by moving through non-dominated solutions that are at a maximum distance of k.

Several statistics can then be collected for several values of k. Examples are the number of clusters and the percentage of the total number of solutions in clusters. Note that for these statistics to be interesting for algorithms like those of [1,6,9,12], resulting *singleton graphs* are not taken into account. In this paper, we collect these statistics for various instance classes of the BTSP and the BQAP. For both problems, solutions are represented as permutations and usual k-exchange neighborhoods can be applied. In order to judge the level of clustering present in the various instance classes, we compare the resulting statistics to those that would be observed if the solutions were distributed uniformly at random in the search space. To do so, we generated 10,000 random permutations for each instance size and measured the resulting minimum, mode, and maximum of the distribution of distance values (pairwise distances measured among the generated solutions); if the number of observed clusters drops down near the mode, this would suggest that the approximate solutions are scattered randomly in the solution space, which, intuitively, can be seen as a disadvantage for the local search algorithms mentioned above.

2.1 The Biobjective TSP

In the biobjective Traveling Salesman Problem (BTSP) are given n cities and a bidimensional distance vector for each pair of cities. The goal is to find a set of Hamiltonian tours that is minimal with respect to the notion of Pareto optimality. This problem is known to be NP-hard since its single objective version is also NP-hard [14].

In the single objective version case, most local search algorithms are based on the k-edge-exchange neighborhoods, where two solutions are neighbored if they differ in at most k edges. Hence, we base our distance measure also on these neighborhoods. Unfortunately, for these neighborhoods the exact distance cannot be computed efficiently. For example, in the case of the 2-edge-exchange neighborhood the problem of determining the minimum distance between two solutions corresponds to the problem of *sorting a permutation by the minimum number of reversals*, which is known to be NP-hard [4]. Therefore, we use the number of different edges between tours, which is an approximation of the exact distance between tours; for example, in the case of the 2-edge-exchange neighborhood, this measure is in the worst case twice the minimum distance.

The BTSP studied in this article is defined by two distance matrices. Here, we examined BTSP instances of 100, 300 and 500 cities where we have chosen the two distance matrices in order to have three different classes of instances: (1) both are random Euclidean matrices (`Eucl`), (2) both are random distance matrices (`RDM`), and (3) one is a random Euclidean and the other is a random distance matrix (`Mix`).*

The plot of Fig. 1 shows the average number of non-dominated solutions found for each type and size of instance. Interestingly, the results indicate an increasing

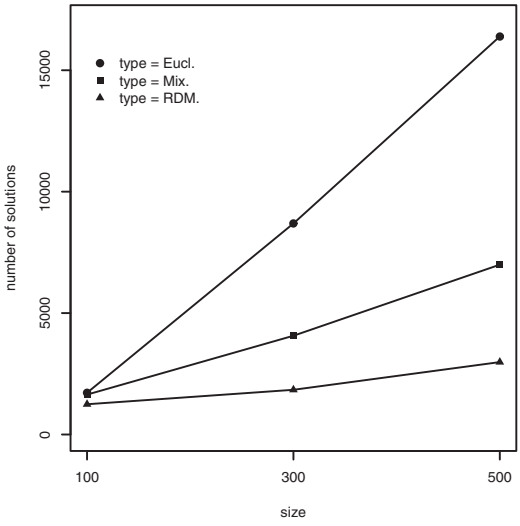


Fig. 1 Average number of non-dominated solutions in the BTSP instances

* See [7] for a detailed explanation about the different kinds of TSP instances.

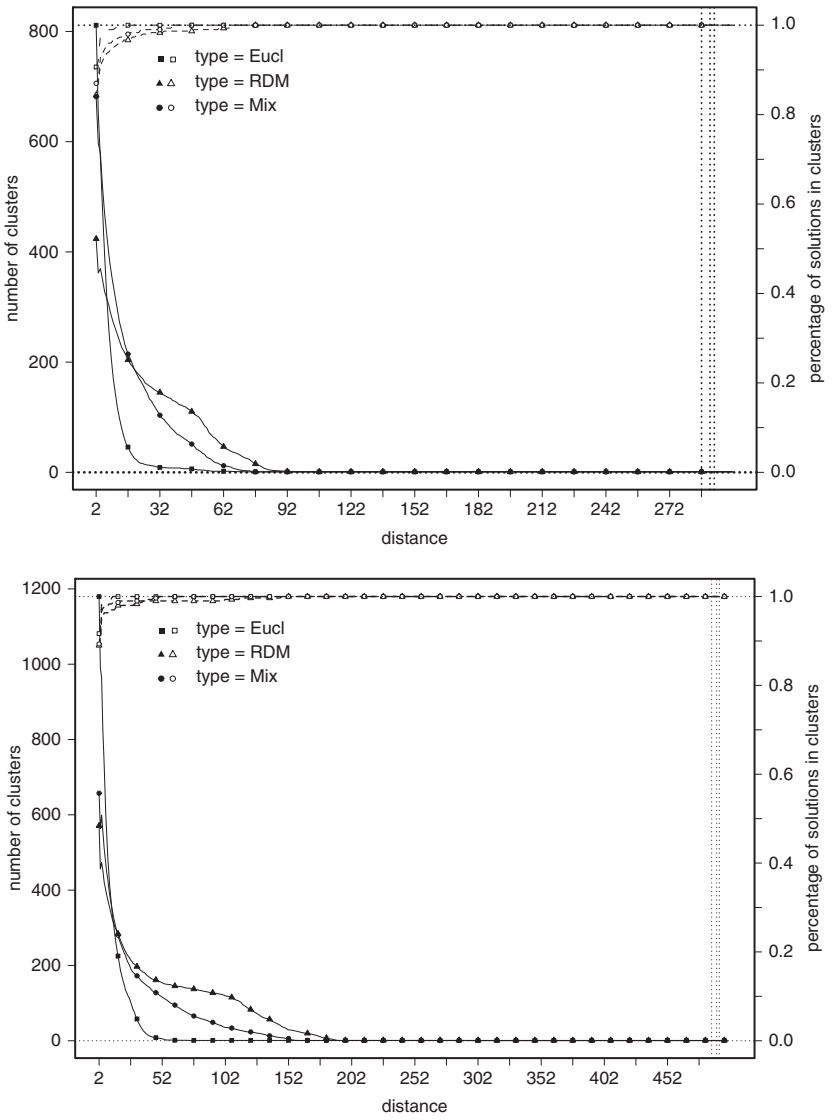


Fig. 2 Number of clusters and solutions in clusters for two BTSP instances

difference between the number of solution produced in `Eucl` and `RDM` instances as the instance size grows.

Some results of the analysis are presented in the plots of Fig. 2. Given are the average number of clusters on the left axis, represented by solid lines and filled

symbols, and on the right axis the average percentage of the total number of solutions in clusters, represented by dashed lines and empty symbols, for an instance of size 300 (top plot) and one of size 500 (bottom plot); the results on the instances with 100 cities are very similar. The squares, triangles and circles indicate `Eucl`, `RDM` and `Mix` instances, respectively. In addition, we indicate the minimum, the mode and the maximum distance found among all pairs of random permutations by the first, second and third dashed vertical line in the plot (visible at the right of each plot), respectively.

The most striking observation is the difference between the range of distance values between random permutations and those resulting from the BTSP instances; while almost all solutions of the latter are concentrated in large clusters with less than one third of the maximum distance, for random permutations the minimum distance between any two tours is about 95% of the maximally possible distance, as indicated by the leftmost vertical line. Hence, we can clearly conclude that approximate solutions to these BTSP instances are strongly clustered. Interestingly, we also see for small values of k some differences in the number of clusters among the three instance types. The tendency is that there is a lower number of clusters in `RDM` instances than in the other types of instances; this effect could be due to the lower number of solutions we have found in `RDM` instances.

2.2 The Biobjective QAP

The quadratic assignment problem is commonly described as an assignment of a set of n facilities to n locations such that the sum of the products of the flow among each pair of facilities by the distance between their assigned locations is minimized. Here we took the biobjective model introduced in [8] which consists of two flow matrices but always keeps the same distance matrix.

In the k-exchange neighborhoods for the BQAP, two solutions are neighbored if they differ in the assignment of at most k facilities to locations. The distance considered between each pair of observed solutions in the BQAP case is the minimum number of swaps of two assignments of a facility to a location, which can be computed efficiently [2, 13]. We generated instances of three different sizes (25, 50 and 75 locations/facilities) and three different values for the correlation between flow matrices (0.75, 0.0, -0.75), which translates into certain correlations between the objectives. In addition, unstructured and structured distance and flow matrices were considered; the instance generator for structured instances imposes clusters between locations on the distance matrix and null entries on the flow matrices. Additionally, the non-zero entries of the flow matrices show a strongly skewed distribution, with high values occurring rarely. In the unstructured instances, the entries of the distance and the two flow matrices are generated according to a uniform distribution in the range $[0, 99]$.

The average number of non-dominated solutions found for each correlation value and size of instances are given in the two plots of Fig. 3 for unstructured (top) and

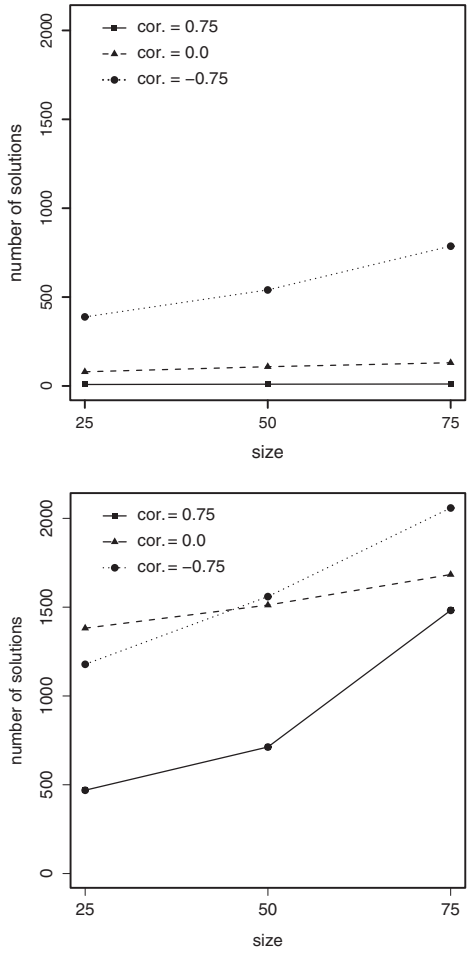


Fig. 3 Average number of non-dominated solutions in unstructured (top) and structured (bottom) BQAP instances

structured (bottom) instances. Note that the correlation has a strong effect on the number of solutions for unstructured instances, that is, the lower the correlation, the larger is the number of solutions returned. This observation is consistent with other results in the literature, such as in [10] for the multiobjective shortest path problem. However, this effect is weaker in structured instances, which might be due to the presence of structure in the input data. Interestingly, the number of solutions obtained in structured instances is larger than those in unstructured instances.

The results of our analysis are given in the two plots of Fig. 4 for unstructured (top) and structured (bottom) instances of size 50; the information given is analogous to the BTSP case. The geometric symbols indicate different degrees of correlation between the flow matrices. Once again, the observations are similar for different sizes.

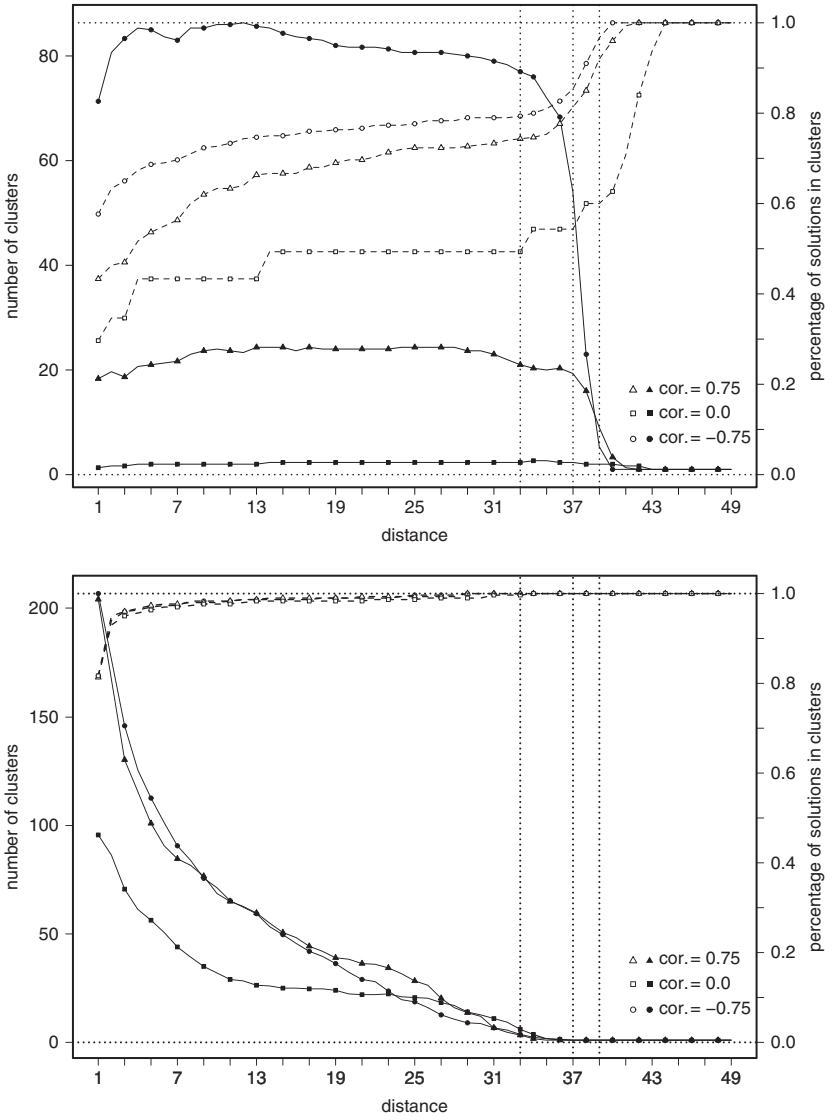


Fig. 4 Number of clusters and solutions in clusters for unstructured (*top*) and structured BQAP (*bottom*) instances

The curves indicate a very different pattern from those observed for the BTSP. For instance, the degree of clustering of approximate solutions is much smaller for the BQAP instances than for the BTSP instances, as it is indicated by the much lower percentage of solutions in clusters and the much slower decrease of the number of clusters. Moreover, the differences between the unstructured and structured BQAP instances are rather large. While for the structured BQAP instances, the shapes of the

curves have some similarities to those for the BTSP, for the unstructured instances the very strong decrease in the number of clusters within the range of distances found between random permutations (most clearly visible for the instances with 0.75 and 0.0 correlation) is striking. In fact, this means that for unstructured instances there are approximate solutions that are quite far away from each other with respect to the neighborhood chosen. In addition, especially on the unstructured instances, the correlation between the flow matrices strongly affects the number of clusters.

3 Discussions and Conclusions

The experimental results obtained by Borges and Hansen [3] on the BTSP, though for a more limited number and type of instances, match the results observed in this article: approximate solutions for this problem are typically clustered with respect to the well-known 2-edge-exchange neighborhood. Differently, the results obtained for the BQAP indicate that the structure of the input data plays an important role on the degree of clustering of approximate solutions. In fact, several statistics on the input data, such as the flow dominance, change considerably with the presence or lack of structure [8]. Therefore, it is expected that the level of clustering might change as well.

This analysis also suggests that stochastic local search algorithms should use different general-purpose search strategies for different problems and for different instance features. For instance, the number of solutions collected for Euclidean BTSP instances and for structured BQAP instances indicates that local search algorithms for those problems should forcedly use an archive of bounded size.

Moreover, exploring extensively non-dominated solutions close to approximate solutions for scalarizations of the objective function vector, as performed in [1, 6, 9, 12], seems to be more appropriate for the BTSP under the 2-exchange neighborhood than for the BQAP. In fact, these results explain the good performance of the simple approach proposed in [12] on benchmark instances of the BTSP.

Finally, we remark that, despite the theoretical interest of this topic, the type of analysis reported in this article helps algorithm designers in identifying the algorithmic strategies that should be used if the real-life input data has certain features. For the specific case of the BQAP, both the correlation between flow matrices and the structure of the input data can be easily detected. Similarly, Euclidean distances are easy to detect for the case of the BTSP. In many cases, the instance features will arise almost naturally from the problem formulation. For example, if the BQAP arises from the task of allocating facilities to locations in a new building for a hospital, where the flow of doctors and the flow of nurses between facilities must be taken into account, one should expect that there should be a positive correlation between the two flow matrices. An example for the BTSP consists on planning a sightseeing tour where both distance and cost are two objectives to minimize; the distance matrix is Euclidean whereas the cost matrix is usually not.

Acknowledgements Thomas Stützle acknowledges support of the Belgian F.R.S.–FNRS, of which he is a research associate.

References

1. Andersen K, Jörnsten K, Lind M (1996) On bicriterion minimal spanning trees: an approximation. Comput Oper Res 23(12):1171–1182
2. Bachelet V (1999) Métaheuristiques paralleles hybrides: Application au probleme d'affectation quadratique. Dissertation, Université des Sciences et Technologies de Lille, France
3. Borges PC, Hansen MP (2000) A study of global convexity for a multiple objective travelling salesman problem. In: Ribeiro CC, Hansen P (eds) Essays and surveys in metaheuristics. Kluwer, Dorderecht, pp 129–150
4. Caprara A (1997) Sorting by reversals is difficult. In: Proceedings of the first annual international conference on computational molecular biology. ACM, New York, NY, pp 75–83
5. Ehrgott M, Klamroth K (1997) Connectedness of efficient solutions in multi criteria combinatorial optimization. Eur J Oper Res 97:159–166
6. Hamacher HV, Ruhe G (1994) On spanning tree problems with multiple objectives. Ann Oper Res 52:209–230
7. Johnson DS, McGeoch LA (1997) The travelling salesman problem: a case study in local optimization. In: Aarts EHL, Lenstra JK (eds) Local search in combinatorial optimization. Wiley, Chichester, UK, pp 215–310
8. Knowles J, Corne D (2002) Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem. In: Abraham A, Ruiz del Solar J, Koppen M (eds) Soft computing systems: design, management and applications. IOS, Amsterdam, The Netherlands, pp 271–279
9. Morita H, Gandibleux X, Katoh N (2001) Experimental feedback on biobjective permutation scheduling problems solved with a population heuristic. Found Comput Decision Sci 26(1):23–50
10. Mote J, Murthy I, Olson DL (1991) A parametric approach to solving bicriterion shortest path problem. Eur J Oper Res 53:81–92
11. Paquete L (2005) Stochastic local search algorithms for multiobjective combinatorial optimization: methods and analysis. Dissertation, FB Informatik, TU Darmstadt, Germany
12. Paquete L, Stützle T (2003) A two-phase local search for the biobjective traveling salesman problem. In: Fonseca CM et al. (eds) Evolutionary multi-criterion optimization (EMO 2003), vol 2632, Lecture Notes in Computer Science. Springer, Berlin, Germany, pp 479–493
13. Schiavinotto T, Stützle T (2007) A review of metrics on permutations for search space analysis. Comput Oper Res 34(10):3143–3153
14. Serafini P (1986) Some considerations about computational complexity for multiobjective combinatorial problems. In: Jahn J, Krabs W (eds) Recent advances and historical development of vector optimization, vol 294, Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, pp 222–231

Computational Results for Four Exact Methods to Solve the Three-Objective Assignment Problem

Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott

Abstract Most of the published exact methods for solving multi-objective combinatorial optimization problems implicitely use properties of the bi-objective case and cannot easily be generalized to more than two objectives. Papers that deal explicitely with three (or more) objectives are relatively rare and often recent. Very few experimental results are known for these methods and no comparison has been done. We have recently developed a generalization of the two phase method that we have applied to the three-objective assignment problem. In order to evaluate the performance of our method we have implemented three exact methods found in the literature. We provide an analysis of the performance of each method and explain the main difficulties observed in their application to the three-objective assignment problem.

Keywords: Assignment problem · Computational results · Exact methods · Multi-objective combinatorial optimization

1 The Multi-Objective Assignment Problem

Efficient algorithms to solve the single-objective assignment problem are well known. In this paper we consider the assignment problem with p objectives (pAP). It can be formulated as follows:

$$\text{“min”}(z_1(X),\ldots,z_p(X)) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^k x_{ij} \qquad k=1,\ldots,p,$$
$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i=1,\ldots,n,$$

A. Przybylski (✉)
Laboratoire d'Informatique de Nantes Atlantique, FRE CNRS 2729 – Université de Nantes. 2, rue de la Houssinière, BP92208, F 44322 Nantes Cedex 03 – France,
e-mail: anthony.przybylski@univ-nantes.fr

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

$$\begin{aligned} &\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,\dots,n, \\ &x_{ij} \in \{0,1\} \qquad i,j = 1,\dots,n, \end{aligned} \tag{pAP}$$

where all objective function coefficients c_{ij}^k are non-negative integers and $x = (x_{11},\dots,x_{nn})$ is the matrix of decision variables.

Let X denote the set of feasible solutions of (pAP). We call $\mathbb{R}^{n^2}, X \subseteq \{0,1\}^{n^2} \subseteq \mathbb{R}^{n^2}$, decision space and $\mathbb{R}^p, Y = \{z(x) : x \in X\} \subseteq \mathbb{N}^p \subseteq \mathbb{R}^p$, objective space. Y is also called the outcome set.

In the following, for $y^1, y^2 \in \mathbb{R}^p$ we shall write $y^1 \leqq y^2$ if $y_k^1 \leq y_k^2, k = 1,\dots,p$, $y^1 \leq y^2$ if $y^1 \leqq y^2$ and $y^1 \neq y^2$, and $y^1 < y^2$ if $y_k^1 < y_k^2, k = 1,\dots,p$. We shall use $\mathbb{R}^p_{\leqq} := \{x \in \mathbb{R}^p : x \geqq 0\}$ and analogously $\mathbb{R}^p_{\leq}$ and $\mathbb{R}^p_{<}$.

In multi-objective optimization there is in general no feasible solution which minimizes all objectives simultaneously.

Definition 1. A feasible solution $x^* \in X$ is called *efficient (weakly efficient)* if there does not exist any other feasible solution $x \in X$ such that $z(x) \leq z(x^*)$ ($z(x) < z(x^*)$). $z(x^*)$ is then called a *nondominated point (weakly nondominated point)*. If $x, x' \in X$ are such that $z(x) \leq z(x')$ ($z(x) < z(x')$), we say that x dominates x' (x strictly dominates x') and $z(x)$ dominates $z(x')$ ($z(x)$ strictly dominates $z(x')$).

Definition 2. Supported efficient solutions are optimal solutions of a weighted sum single objective problem $\min\{\lambda_1 z_1(x) + \dots + \lambda_p z_p(x) : x \in X\}$ for some $\lambda \in \mathbb{R}^p_{<}$. Their image in objective space are called supported nondominated points.

We use the following notations (definitions can be found in [9]):

- Y_N: the set of nondominated points,
- Y_{SN}: the set of supported nondominated points,
- Y_{NN}: the set of nonsupported nondominated points,
- $X_E, z(X_E) = Y_N$: a complete set of efficient solutions,
- $X_{SE}, z(X_{SE}) = Y_{SN}$: a complete set of supported efficient solutions,
- $X_{NE}, z(X_{NE}) = Y_{NN}$: a complete set of nonsupported efficient solutions.

2 Experimental Context

We have solved (3AP) in the sense of determining a complete set X_E, i.e., to find at least one efficient solution for every nondominated point using the methods described in [4, 7–9]. We have generated a series of 10 instances of size varying from 5×5 to 50×50 with a step of 5. The objective function coefficients are integers generated randomly in $(0, 20)$. A computer with a P4 EE 3.73 Ghz processor and 4 Gb of RAM has been used for the experiments. All methods have been implemented in C. The binaries have been obtained using the compiler gcc with optimizer option -O3. We have used CPLEX 9.1 to solve the (single objective) assignment problems

with some additional constraints that need to be solved in the methods of [4,7,8]. In the next sections, a brief description of each method used is given with an analysis of the experimental results.

3 The Method of Sylva and Crema (2004)

Given a vector $\lambda \in \mathbb{R}^p_>$, this method computes the nondominated points (with one corresponding efficient solution for each point) in increasing order of $\lambda^T z$. This is done by solving one subproblem for each new efficient solution. After finding k nondominated points $y^1, \ldots, y^k$, the following subproblem is solved

$$\min \left\{ \lambda^T z(x) : x \in X \setminus \bigcup_{i=1}^{k} D^i \right\},$$

where $D^i = \{x \in X : y^i \leqq z(x)\}$. There are two possible cases:

- The subproblem is feasible so that the next nondominated point y^{k+1} is found (with a corresponding solution) and the algorithm proceeds to solve the next subproblem.
- The subproblem is infeasible. Then $\{y^1, \ldots, y^k\} = Y_N$ and the algorithm ends.

Let $x \in X$ with $y = z(x)$. To formulate the constraint $x \notin D^i$, Sylva and Crema [4] propose to use disjunctive constraints

$$x \notin D^i \Longleftrightarrow y_1 \leq y^i_1 - 1 \text{ or } y_2 \leq y^i_2 - 1 \text{ or } \ldots \text{ or } y_p \leq y^i_p - 1$$

based on the integrality of the costs. These constraints are linearized in the usual way:

$$x \notin D^i \Longleftrightarrow \begin{cases} y_k \leq (y^i_k - 1)x^i_k + M(1 - x^i_k) \text{ for } k = 1, \ldots, p \\ \sum_{k=1}^{p} x^i_k \geq 1 \\ x^i_k \in \{0,1\} \text{ for } k = 1, \ldots, p, \end{cases}$$

where M is a big number (an appropriate value can be obtained using the range of data and the size of the problem).

Therefore, after each iteration $p+1$ constraints and p new binary variables are added to the problem. Consequently, the subproblems to be solved become larger and larger and more and more constrained.

With this method, we have solved the 5×5 instance in 0.15 s ($|Y_N| = 12$) and the 10×10 instance in 99,865 s ($|Y_N| = 221$). We have not been able to solve larger instances with this method. The main limitation is clearly the huge difficulty for CPLEX to solve the subproblems after some iterations. This fact is illustrated in Fig. 1. It is clearly not possible with this method to solve problems with large cardinality of Y_N.

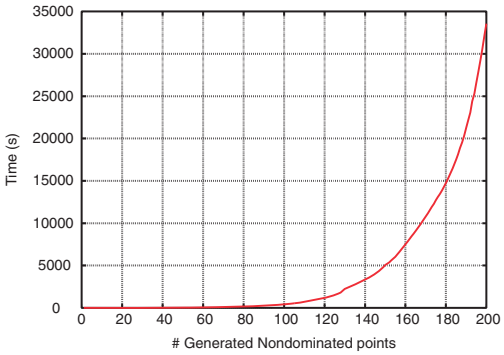


Fig. 1 Number of nondominated points generated versus time on the 10×10 instance with the method of [4]

4 The Method of Tenfelde–Podehl (2003)

This method performs three steps:

Step 1: Let $P(i)$ be the problem with $p-1$ objectives defined by

$$\min_{x \in X}(z_1(x), \ldots, z_{i-1}(x), z_{i+1}(x), \ldots, z_p(x)).$$

For all $i \in \{1, \ldots, p\}$, $P(i)$ is solved. This gives nondominated points, and according to Theorem 4 in [2] the ideal point and the nadir point.

Step 2: The nondominated points found in Step 1 are used to determine boxes in objective space.

Step 3: The boxes defined in Step 2 are explored by dichotomy.

We only describe Step 2. At the end of Step 1, we know that all nondominated points are located in the large box defined by the ideal and the nadir point. However, we already know some nondominated points inside this box and it is therefore not necessary to explore the whole box. For all y found in Step 1, Tenfelde–Podehl [7] proposes to introduce p hyperplanes $h^i(y) = \{y' \in \mathbb{R}^p : y'_i = y_i\}$. These hyperplanes are used to divide the initial box into small boxes (Fig. 2).

Next, small boxes where no nondominated points can exist are excluded using the following properties:

Property 1. Let P be a problem with p objectives and let y^* be a nondominated point obtained in Step 1. The following statements hold.

(a) Every feasible point y with $y^* \leq y$ is dominated.
(b) There does not exist a feasible point y for which $y \leq y^*$.

Property 2. Let P be a problem with p objectives and let y^* be a nondominated point found in Step 1, obtained for problem $P(i)$. Then there does not exist a feasible point y for which

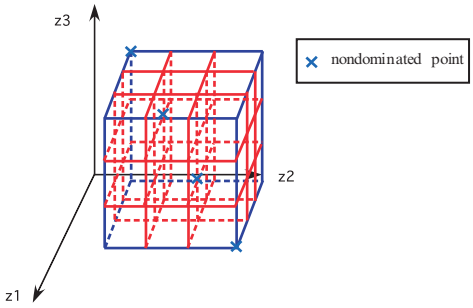


Fig. 2 Illustration of the division of the box defined by the ideal and the nadir point with three objectives

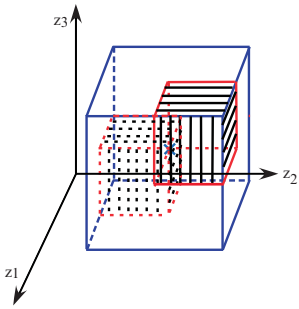


Fig. 3 Illustration of Property 1 with 3 objectives

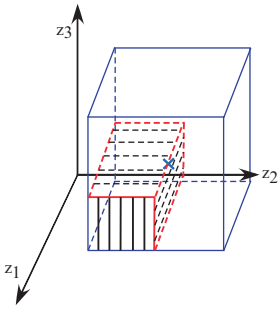


Fig. 4 Illustration of Property 2 with 3 objectives and a nondominated point obtained by $P(1)$

$$(y_1,\ldots,y_{i-1},y_{i+1},\ldots,y_p) \leq (y_1^*,\ldots,y_{i-1}^*,y_{i+1}^*,\ldots,y_p^*).$$

The properties are illustrated in Figs. 3 and 4. Small boxes included in the hashed areas are excluded.

Then it remains to determine boxes where nondominated points not found in Step 1 may exist. This is done using Property 3.

Property 3. Let P be a problem with p objectives and let y^* be a nondominated point not found in Step 1. For $i \in \{1,\ldots,p\}$, there exists a nondominated point y^i for $P(i)$ such that $y_i^i > y_i^*$ and $(y_1^i,\ldots,y_{i-1}^i,y_{i+1}^i,\ldots,y_p^i) \leq (y_1^*,\ldots,y_{i-1}^*,y_{i+1}^*,\ldots,y_p^*)$.

Using one nondominated point of each problem $P(i)$, Property 3 yields an intersection of p boxes (which is possibly empty or the union of small boxes) called

Table 1 Experimental results with the method of [7]

Instance size	Remaining intervals	Total inclusions	Memory required	CPU time (s)
5	40	196	1.5 Kb	0.04
10	6,849	1,563,082	11.9 Mb	97.30
15	23,531	25,902,170	197.6 Mb	544.53
20	213,414	2,238,368,109	16.7 Gb	×

admissible interval. All combinations consisting of one nondominated point of each problem $P(i)$ must be used to determine these boxes. All small boxes that are not included in an admissible interval are excluded. The remaining small boxes are called *remaining intervals*.

Finally, it remains to explore the remaining intervals. Because their union is not convex and there are too many remaining intervals, it is necessary to group them such that ideally:

(a) The grouping yields a convex box
(b) Each of the remaining intervals is explored once
(c) The smallest possible number of admissible intervals has to be searched

This is a difficult problem. Tenfelde–Podehl [7] proposes to use a heuristic (that ensures condition (a) and (b) above but not necessarily condition (c)). This heuristic uses properties of the location of remaining intervals inside the admissible intervals. Therefore, it is necessary to first determine the sets of admissible intervals and remaining intervals, and the set of remaining intervals included in each admissible interval.

Table 1 summarizes the experimental results we observed. The column "Total inclusions" means the sum of the number of remaining intervals in each admissible interval. As we can see, this number becomes huge quickly. For the 20×20 instance, 16.7 Gb of memory are required. Thus, the limitation of this method is clearly its memory requirement.

5 The Method of Laumanns et al. (2005)

This is a method based on the ε-constraint approach [3]:

$$\begin{aligned} &\min z_1(x) \\ &\text{subject to } z_k(x) \leq \varepsilon_k \qquad k = 2, \ldots, p \\ &\qquad\qquad x \in X. \end{aligned} \tag{$P(\varepsilon)$}$$

For a given ε, if $P(\varepsilon)$ is feasible, then the optimal solution is weakly efficient. All efficient solutions are optimal solutions of problems $P(\varepsilon)$ with appropriate choices of ε. While the adaptive variation of ε to find all nondominated points

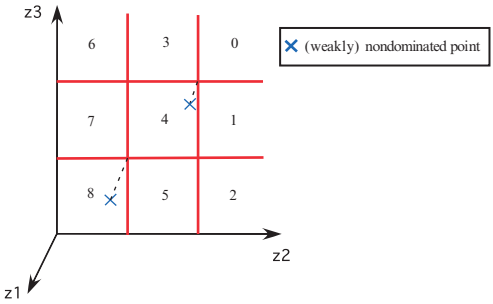


Fig. 5 Illustration of the hypergrid with three objectives

is straightforward with two objectives, it has long remained unsolved with three objectives and more. Laumanns et al. [8] have proposed a generalization and have provided a source code as example (application to the three-objective multidimensional binary knapsack problem, available at `http://www.tik.ee.ethz.ch/laumanns`). In the following, we give a brief description of the implementation done by the authors of the method which is slightly different from the description in their paper [8] using a lexicographic objective.

This method uses all (weakly) nondominated points already found. This is done using a $(p-1)$ dimensional hypergrid, which partitions the whole objective space into hyperrectangles parallel to the axis of objectives z_2 to z_p (Fig. 5). The hyperrectangles are explored by decreasing order of the cell index by defining problems $P(\varepsilon)$ with ε equal to the upper-right corner of each hyperrectangle. In the application to $(3AP)$, we can subtract 1 from each component, because we assume costs to be integer.

Solving a problem $P(\varepsilon)$ two cases can occur:

- If there is no feasible solution, or a point already found or dominated by a point already found is obtained then the hyperrectangle is empty (in the sense that there is no nondominated point inside).
- Otherwise, a new (weakly) nondominated point is found, the exploration of the grid is stopped, a new grid is defined with this new point, and the exploration of this new grid is started. It is not necessary to explore all hyperrectangles in this new grid, since it is already known that some hyperrectangles are empty.

The experimental results of this method on (3AP) are shown in Table 2. The time needed by this method increases exponentially, which is due to the huge number of nondominated points. Indeed, with the number of (weakly) nondominated points found during the exploration, the grid becomes thinner and thinner. Consequently, it is necessary to solve more and more subproblems to find a new (weakly) nondominated point. This fact is illustrated in Fig. 6.

Table 2 Experimental results for the methods of [4], [7], [8] and [9] on $(3AP)$ instances with CPU time in seconds.

Instance size	$\#Y_N$	[4]	[7]	[8]	[9]
5	12	0.15	0.04	0.15	0.00
10	221	99,865.00	97.30	41.70	0.08
15	483	×	544.53	172.29	0.36
20	1,942	×	×	1,607.92	4.51
25	3,750	×	×	5,218.00	30.13
30	5,195	×	×	15,579.00	55.87
35	10,498	×	×	101,751.00	109.96
40	14,733	×	×	×	229.05
45	23,941	×	×	×	471.60
50	29,193	×	×	×	802.68

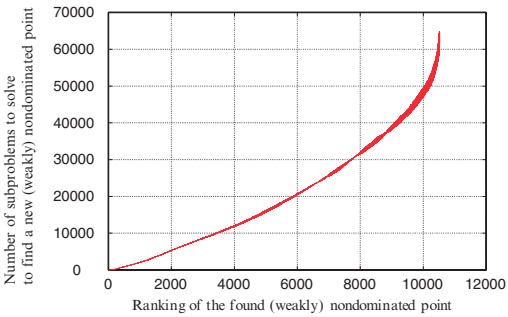


Fig. 6 Number of problems to be solved for finding one new (weakly) nondominated on the 35×35 instance

6 The Generalization of the Two Phase Method

The two phase method [10] is an exact method to solve bi-objective combinatorial optimization problems. Its main idea is to make intensive use of efficient algorithms for the single objective case. However, as these efficient algorithms are problem specific, it is not possible to modify the problem structure by adding constraints and only the weighted sum of the objectives can be used. In Phase 1, supported solutions are determined with a dichotomic scheme using Geoffrion's theorem. In Phase 2, a search area where nonsupported points may exist is determined using the supported points. Then this search area is explored with a problem specific procedure. Very often, to ensure complete enumeration of the search area, bands are determined. The determination of the supported solutions in Phase 1 and of the search area in Phase 2 depends on the natural order of nondominated points in the bi-objective case.

In a previous work, we have proposed an algorithm for the computation of a set X_{SE} of a multi-objective linear integer problem [5]. This algorithm requires to be able to efficiently solve the single objective problem. Therefore it can be used for a generalization of Phase 1.

In order to determine a set X_{NE} in Phase 2 it is necessary to describe the search area where nonsupported points may be found. However, for $p > 2$ objectives, the search area cannot be described by simple geometric constructions defined by consecutive supported solutions. A first description can be given using lower and upper bound sets as defined in [1].

$L = (\operatorname{conv} Y_{SN})_N$ is a lower bound set for Y_N, i.e. $Y_N \subseteq L + \mathbb{R}^p_{\leqq}$. Any set of feasible points U that does not contain y^i, y^j with $y^i \leq y^j$ or $y^j \leq y^i$ is an upper bound set for Y_N, i.e. $Y_N \cap (U + \mathbb{R}^p_{>}) = \emptyset$. In particular in the context of the two phase method, we can use $U = Y_{SN}$. Both L and U are obtained in Phase 1. Then the search area can be defined by $(L + \mathbb{R}^p_{\leqq}) \setminus (U + \mathbb{R}^p_{>})$.

However, with this description, it is difficult to define bands to be explored in the search area. In the bi-objective case, if we are letting $Y_{SN} = \{y^1, \ldots, y^r\}$ such that $y^1_1 < y^2_1 < \ldots < y^r_1$ (and therefore $y^1_2 > y^2_2 > \ldots > y^r_2$), the search area consists of triangles with corner points y^i, y^{i+1} and the local nadir point $\eta^i = (y^{i+1}_1, y^i_2)$, $i = 2, \ldots, r-1$. So, in the bi-objective case, an equivalent way to define the search area is $(L + \mathbb{R}^2_{\leqq}) \cap \bigcup_{i=1}^{r-1} (\eta^i - \mathbb{R}^2_{\leqq})$. This way to describe the search area is more convenient for the application of the two phase method. Indeed, in order to explore the search area, bands containing the whole triangles are explored. The slimmest possible bands for this are defined by the hypothenuse of the triangle and the parallel line containing η_i. Therefore we propose to describe the search area in an equivalent way, in order to be able to define the slimmest bands to explore it. For this, by analogy with the bi-objective case, we determine a set of points $D(U) \subseteq \mathbb{R}^p$ such that $(L + \mathbb{R}^p_{\leqq}) \setminus (U + \mathbb{R}^p_{>}) = (L + \mathbb{R}^p_{\leqq}) \cap \bigcup_{u \in D(U)} (u - \mathbb{R}^p_{\leqq})$. The details of the characterisation of $D(U)$ and its computation, as well as the exploration of bands by a ranking method can be found in [9].

Table 2 summarizes all experiments for the four methods. We can see that the two phase method performs much faster than the other methods which is not really surprising according to our results for (2AP) in [6]. For $(3AP)$, the gap in computational time is larger. For example, the instance 35×35 is solved almost 1000 times faster with the two phase method than with the method of [8] and the gap is growing rapidly!

7 Conclusion

The difficulties to solve $(3AP)$ instances with the methods of [4], [7] and [8] are caused by the large cardinality of Y_N. It causes the difficulty to solve the subproblems with the method of [4], the huge memory requirement for the method of [7], and the huge number of subproblems to be solved with the method of [8]. Consequently, only small or medium size instances can be solved with these methods.

On the other hand, the instances are solved easily with the two phase method. This highlights the importance of exploiting the problem structure when the number of objectives is increasing.

References

1. Ehrgott M, Gandibleux X (2007) Bound sets for biobjective combinatorial optimization problems. Comput Oper Res 34(9):2674–2694
2. Ehrgott M, Tenfelde-Podehl D (2003) Computation of ideal and nadir values and implications for their use in MCDM methods. Eur J Oper Res 151:119–139
3. Haimes Y, Lasdon L, Wismer D (1971) On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Transa Syst Man Cybern 1:296–297
4. Sylva J, Crema A (2004) A method for finding the set of non-dominated vectors for multiple objective integer linear programs. Eur J Oper Res 158(1):46–55
5. Przybylski A, Gandibleux X, Ehrgott M (2007) Two Recursive algorithms for finding all nondominated extreme points in the outcome set of a multi-objective integer programme. Technical Report LINA. Submitted to INFORMS Jounal On Computing
6. Przybylski A, Gandibleux X, Ehrgott M (2008) Two phase algorithms for the bi-objective assignment problem. Eur J Oper Res 185(2):509–533
7. Tenfelde-Podehl D (2003) A recursive algorithm for multiobjective combinatorial problems with Q criteria. Technical Report, University of Graz, Austria
8. Laumanns M, Thiele L, Zitzler E (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. Eur J Oper Res 169(3):932–942
9. Przybylski A, Gandibleux X, Ehrgott M (2007) A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. Technical Report LINA
10. Ulungu EL, Teghem J (1995) The two-phases method: an efficient procedure to solve biobjective combinatorial optimization problems. Found Comput Decision Sci 20(2):149–165

Constraint Optimization Techniques for Exact Multi-Objective Optimization

Emma Rollon and Javier Larrosa

Abstract *MultiObjective Branch and Bound* search has not been widely studied in the multiobjective context. The main reason is the lack of general approximation algorithms to compute lower bound sets. However, many lower bound techniques has been proposed for mono-objective optimization in the *constraint programming* field. In particular, *Mini-Bucket Elimination* (MBE) is a powerful mechanism for lower bound computation. Recently, MBE has been extended to multi-objective optimization problems. The new algorithm, called MO-MBE, computes a lower bound set of the efficient frontier of the problem. We show how to embed MO-MBE in a multi-objective branch and bound search, and we empirically demonstrate the performance of the new approach in two different domains.

Keywords: Constraint programming · Multi-objective branch-and-bound search · Multi-objective lower bounds

1 Introduction

In *Constraint Optimization Problems* (COP) the task is to find the best solution according to some preferences expressed by means of cost functions [1]. *Branch and Bound* (BB) [2] is an exact general search algorithm for COPs solving. BB is the most usual algorithm in the mono-objective case. The efficiency of BB depends on its hability to detect dead-ends, that is, nodes that do not have any solution below. Dead-end detection is done with a heuristic function that computes an underestimation or *lower bound* of the current subproblem. In recent years, many heuristic functions have been proposed. For instance, all *weighted* CSP local consistencies [3, 4] can be used for this purpose. Another alternative is *Mini-bucket*

E. Rollon (✉)
Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edificio Omega – 08034 Barcelona, Spain
e-mail: erollon@lsi.upc.edu

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

elimination (MBE) [5]. MBE is a generic inference method well-known in *Constraint Programming* [6]. It computes a lower bound of the optimal solution (assuming minimization). Therefore, MBE is usually used inside BB to improve its pruning capability.

Many real world problems involve multiple measures of performance, or objectives, which should be optimized simultaneously. The simultaneous optimization of multiple, possibly competing, objective functions deviates from single function optimization in that it seldom admits a single, perfect solution. Instead, multiobjective constraint optimization problems tend to be characterized by a family of alternatives which must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others.

In *MultiObjective Constraint Optimization Problems* the task is to find the *efficient frontier*, that is, the set of equivalent or non-dominated costs of the set of feasible solutions. *MultiObjective Branch and Bound* search (MO-BB) has not been widely studied in the multiobjective context [7]. One reason is the lack of general approximation algorithms to compute lower bounds. In our recent work [8], we have extended MBE from mono-objective to multi-objective optimization problems, yielding *MultiObjective Mini-Bucket Elimination* (MO-MBE). MO-MBE computes a *lower bound set* [9] of the efficient frontier. As a consequence, MO-MBE can be used as a heuristic function in a multiobjective branch and bound algorithm. In this paper we describe how MO-MBE can be combined with multiobjective branch and bound search. The resulting algorithm is a simple, extremely generic, exact multiobjective solving method. Our experiments on *bi-objective combinatorial auctions* and *bi-objective weighted vertex cover* problems demonstrates the performance of the new approach.

The structure of this paper is as follows: Section 2 provides some preliminaries on multiobjective optimization. Section 3 introduces Multiobjective mini-bucket elimination. Section 4 describes the multiobjective extension of a generic branch and bound algorithm and shows how MO-MBE can be integrated as a heuristic function. Section 5 reports some experimental results. Finally, Sect. 6 gives some conclusions and points out some directions of future work.

2 Preliminaries

Let $\mathscr{X} = (x_1, \ldots, x_n)$ be an ordered set of variables and $\mathscr{D} = (D_1, \ldots, D_n)$ an ordered set of domains. Domain D_i is a finite set of potential values for x_i. We call d the largest domain size. The assignment (i.e, instantiation) of variable x_i with $a \in D_i$ is noted $(x_i := a)$. A *tuple* is an ordered set of assignments to different variables $(x_{i_1} := a_{i_1}, \ldots, x_{i_k} := a_{i_k})$. The set of variables $(x_{i_1}, \ldots, x_{i_k})$ assigned by a tuple t, noted $var(t)$, is called its *scope*. The size of $var(t)$ is the *arity* of t. When the scope is clear by the context, we omit the variables and express the tuple as a sequence of domain values $(a_{i_1} \ldots a_{i_k})$. We focus on two basic operations over tuples: the *projection* of t over $A \subseteq var(t)$, noted $t[A]$, is a sub-tuple of t containing only the instantiation

of variables in A. Let t and s be two tuples having the same instantiations to the common variables. Their *join*, noted $t \cdot s$, is a new tuple which contains the assignments of both t and s. Projecting a tuple t over the empty set $t[\emptyset]$ produces the empty tuple λ. We say that a tuple t is a *complete instantiation* when $var(t) = \mathscr{X}$. Sometimes, when we want to emphasize that a tuple is a complete instantiation we will call it X.

Let consider problems with one objective. A *weighted CSP* (WCSP) [10] is a tuple $P = (\mathscr{X}, \mathscr{D}, \mathscr{F}, \top)$, where $\mathscr{X}$ and $\mathscr{D}$ are variables and domains. $\mathscr{F} = \{f_1, \ldots, f_r\}$ is a set of cost functions. Each cost function f_i is defined over $Y_i \subseteq \mathscr{X}$, called its *scope*. f_i associates costs (i.e., numbers) to tuples t such that $var(t) = Y_i$. We make the usual assumption of costs being natural numbers. $\top$ bounds the maximum acceptable cost of solutions. The objective function is, $F(X) = \sum_{i=1}^{r} f_i(Y_i)$. A *solution* is a complete assignment X such that $F(X) < \top$. An *optimal solution* is a solution X such that $\forall X',\ F(X) \leq F(X')$. The *optimum* of the objective function is the value $F(X)$. The task in a WCSP is to find the optimum and one (of the possibly many) optimal solutions X.

Let consider problems with p objectives. $\top = (\top_1, \ldots, \top_p)$ is a vector where each $\top_j \in \mathbb{N}$ is the maximum acceptable cost for the objective j. A p-vector $\mathbf{v} = (v_1, \ldots, v_p)$ is a vector of p components where each $v_j \in \mathbb{N}$ and $v_j \leq \top_j$. Let $\mathbf{v}$ and $\mathbf{u}$ be two *distinct* p-vectors. $\mathbf{v}$ *dominates* $\mathbf{u}$ (noted $\mathbf{v} < \mathbf{u}$) if $\forall j,\ v_j \leq u_j$. The sum of p-vectors is defined as,

$$\mathbf{v} + \mathbf{u} = \begin{cases} \top & \exists j,\ v_j + u_j \geq \top_j \\ (v_1 + u_1, \ldots, v_p + u_p) & otherwise. \end{cases}$$

Let S be a set of p-vectors. We define its *non-domination closure* as $\langle S \rangle = \{\mathbf{v} \in S \mid \forall \mathbf{u} \in S,\ \mathbf{u} \not< \mathbf{v}\}$. Let S_1 and S_2 be two sets closed under non-domination. We say that S_1 *dominates* S_2 (noted $S_1 < S_2$) if $\forall \mathbf{v} \in S_2,\ \exists \mathbf{u} \in S_1\ s.t\ \mathbf{u} < \mathbf{v}$. A p-function f is defined over a set of variables $Y \subseteq \mathscr{X}$ such that $f(Y)$ is a p-vector. Let $x_i \in Y$ and $a \in D_i$, the partial instantiation of f with $x_i := a$, noted $f^{x_i:=a}$, is the new function obtained from f in which x_i has been *fixed* to a. Note that when x_i is the only variable of f, its instantiation produces a constant p-vector.

A *multiobjective weighted constraint satisfaction problem* (MO-WCSP) is defined as $\mathscr{P} = (\mathscr{X}, \mathscr{D}, \mathscr{F}, \top)$, where $\mathscr{X} = \{x_1, \ldots, x_n\}$ and $\mathscr{D} = \{D_1, \ldots, D_n\}$ are variables and domains. $\top$ contains, for each objective, the maximum acceptable cost. $\mathscr{F}$ is a set of p-functions that define the multiobjective function $F(X) = \sum_{f \in \mathscr{F}} f(X)$. Given a complete assignment X, we say that it is *consistent* iff $F(X) \neq \{\top\}$. For clarity reasons, we consider that all the objective functions are additive. However, the same ideas posed in the MO-WCSP framework can be used for modelling problems where the objective functions are, for example, multiplicative (i.e. Probabilistic Frameworks [11]), or a combination of both. A *solution* is a consistent complete assignment. In the constraint programming context, the usual task is to find an optimal solution. A solution X is *efficient* or *Pareto optimal* if there is no better solution (i.e., $\forall X',\ F(X') \not< F(X)$). $\mathscr{X}_E$ is the set of efficient solutions and $\mathscr{EF}$ is the corresponding *efficient frontier*. The task in a MO-WCSP is to compute $\mathscr{EF}$ (and, possibly, one or all efficient solutions for each of its elements).

3 Mini-Bucket Elimination

MultiObjective Mini-Bucket Elimination (MO-MBE) [8] is the extension of the well-known mono-objective approximation algorithm *Mini-Bucket Elimination* (MBE) [5] to the multiobjective context. MO-MBE is a generic approximation algorithm that can be used to bound the efficient frontier when the problem is too difficult to be solved exactly. Assuming minimization problems, MO-MBE provides a lower bound set of the efficient frontier.

In the following, we augment p-functions by letting them to return a non-dominated *set* of p-vectors. MO-MBE uses two operations over p-functions: The *sum* of two p-functions f and g, noted $f+g$, is a new p-function that returns for each tuple the sum of the corresponding p-vectors, previous removal of dominated ones. The *elimination* of variable x_i from p-function f, noted $f \downarrow x_i$, is a new p-function not mentioning x_i that returns for each tuple the best p-vectors with respect to the eliminated variable. Formally, let f and g be two p-functions:

- Their sum $h = f+g$ is defined as,

$$h(t) = \langle \{\mathbf{v} |\ \ t = t' \cdot t'', \mathbf{v} = \mathbf{v}' + \mathbf{v}'', \mathbf{v}' \in f(t'), \mathbf{v}'' \in g(t'')\}\rangle$$

- The elimination of x_i, $h = f \downarrow x_i$ is defined as,

$$h(t) = \langle \{\mathbf{v} \mid \forall a \in D_i,\ \mathbf{v} \in f(t \cdot (x_i := a))\}\rangle$$

Consider as an example the 2-functions f and g in Fig. 1 with $\top = (15,18)$ under domains $\{a,b\}$. The sum $f+g$ is a 2-function $(f+g)(x_1,x_2,x_3)$. Note that in $(f+g)(a,b,a)$, the sum of the 2-vectors $(4,10)$ and $(11,1)$ is $\top$. As $\top$ is dominated by $(10,12)$, it has been removed. The elimination of variable x_3 from $f+g$ is a 2-function $(f+g) \downarrow x_3(x_1,x_2)$. Note that in $(f+g) \downarrow x_3(a,a)$, the 2-vector $(4,9)$ has been removed as a consequence of the non-domination closure. Moreover, $\top$ has also been removed from $(f+g) \downarrow x_3(a,b)$ for the same reason.

f:

x_1	x_2	
a	a	$\{(3,2),(2,8)\}$
a	b	$\{(4,10)\}$
b	a	$\{\top\}$
b	b	$\{\top\}$

g:

x_2	x_3	
a	a	$\{(1,2)\}$
a	b	$\{(2,1)\}$
b	a	$\{(6,2),(11,1)\}$
b	b	$\{\top\}$

$f+g$:

x_1	x_2	x_3	
a	a	a	$\{(4,4),(3,10)\}$
a	a	b	$\{(5,3),(4,9)\}$
a	b	a	$\{(10,12)\}$
a	b	b	$\{\top\}$
b	a	a	$\{\top\}$
b	a	b	$\{\top\}$
b	b	a	$\{\top\}$
b	b	b	$\{\top\}$

$(f+g) \downarrow x_3$:

x_1	x_2	
a	a	$\{(4,4),(3,10),(5,3)\}$
a	b	$\{(10,12)\}$
b	a	$\{\top\}$
b	b	$\{\top\}$

Fig. 1 Sum and projection over 2-functions. $\top = (15,18)$

function MO−MBE($\mathscr{X}, \mathscr{D}, \mathscr{F}, \top, z$)
1. **for each** $i = n \ldots 1$ **do**
2. $\quad \mathscr{B}_i := \{h \in \mathscr{F} \mid x_i \in var(h)\}$;
3. $\quad \{\mathscr{P}_{i_1}, \ldots, \mathscr{P}_{i_r}\}$:=Partition($z, \mathscr{B}_i$);
4. $\quad$ **for each** $k = 1..r$ **do** $g_{i_k} := (\sum_{f \in \mathscr{P}_{i_k}} f) \downarrow x_i$;
5. $\quad \mathscr{F} := (\mathscr{F} \cup \{g_{i_1}, \ldots, g_{i_r}\}) - \mathscr{B}_i$;
6. **endfor**
7. **return** g_1

endfunction

Fig. 2 Description of MO-MBE. The input is a MO-WCSP instance $(\mathscr{X}, \mathscr{D}, \mathscr{F}, \top)$. The output is g_1, a zero-arity p-function which contains a lower bound set of the efficient frontier

MO-MBE (Fig. 2) has a control parameter z. It processes the problem eliminating variables one by one. For each variable x_i, the algorithm computes the so called *bucket* of x_i (line 2), noted $\mathscr{B}_i$, which contains all p-functions in $\mathscr{F}$ having x_i in its scope. Ideally, a new p-function would be computed by summing all functions in $\mathscr{B}_i$ and subsequently eliminating x_i. Since this is very space consumming, the bucket is partitioned into so-called *mini-buckets* (line 3). Each mini-bucket contains p-functions such that they do not jointly mention more than $z+1$ variables. In each mini-bucket the functions are summed and subsequently x_i is eliminated (line 4). Then, $\mathscr{F}$ is updated by removing the functions in $\mathscr{B}_i$ and adding each g_{i_k} (line 5). After the last elimination, only an empty-scope p-function (i.e., a non-dominated set of p-vectors) remains. It contains a lower bound set of the original problem (line 7). Note that, if MO-MBE returns $\{\top\}$ the problem does not have any solution. In general, greater values of z increment the number of p-functions included in each mini-bucket. Therefore, the lower bound set will be presumable closer to the efficient frontier. However, greater values of z produce higher arity functions which require more resources (i.e., space and time).

Theorem 1. *[8] MO-MBE with accuracy parameter z is space* $O(e \times \prod_{j=1}^{p-1} \top_j \times d^{z-1})$ *and time* $O(e \times \prod_{j=1}^{p-1} \top_j^2 \times d^z)$*, where e is the number of p-functions,* $\top_j$ *is the bound of objective j, p is the number of objectives, and d is the largest domain size.*

4 Depth First Branch and Bound

MultiObjective Branch-and-Bound (MO-BB) is a recursive description of a generic search schema for MO-WCSP solving. It searches depth-first the tree defined by the problem. During search, MO-BB maintains a set of non-dominated p-vectors corresponding to the best solutions found so far. In the minimization case, those vectors are an *upper bound set* or *top* of the optimal solution. When a new solution is found, its costs are added to the top and the non-dominated ones are retained as new top. Moreover, for each partial assignment, the algorithm computes a *lower bound set* using a *bounding evaluation function*, that is, an underestimation of its

```
procedure MO-BB(𝒳, 𝒟, ℱ, ⊤, ℰℱ)
1.  if 𝒳 = ∅ then ℰℱ := ⟨ℰℱ ∪ ℱ⟩;
2.  else
3.    x_i := Select(𝒳);
4.    for each a ∈ D_i do
5.      ℱ' := {f(x_i := a) | f ∈ ℱ};
6.      if ℰℱ ⋠ LB(𝒳 − {x_i}, 𝒟 − {D_i}, ℱ', ⊤) then
7.        MO-BB (𝒳 − {x_i}, 𝒟 − {D_i}, ℱ', ⊤, ℰℱ);
8.      endif
9.    endfor
10. endif
endprocedure
```

Fig. 3 Multi-objective depth-first branch and bound for optimization task

efficient frontier that can be obtained in the remaining problem. If the lower bound set is dominated by the top, the current path cannot lead to better solutions and the current branch can be pruned. As a result, the algorithm backtracks to a previous node.

In its description, MO-BB search (Fig. 3) receives a set of variables $\mathscr{X}$ and the set of its feasible values $\mathscr{D}$, a set of p-functions $\mathscr{F}$, a top vector $\top$ and a non-dominated set $\mathscr{EF}$. After an initial call MO-BB $((x_1,\ldots,x_n),(D_1,\ldots,D_n),(f_1,\ldots,f_e),\top,\mathscr{EF}=\{\top\})$, the algorithm returns the efficient frontier of the problem in $\mathscr{EF}$. During search, the current efficient frontier is kept in $\mathscr{EF}$. When no variable remains, the current assignment is one of the best solutions found so far, so the efficient frontier is updated (line 1). Note that when there is no more variable to assign, $\mathscr{F}$ contains an empty scope p-function, that is, a constant p-function containing the optimal p-vectors of the current assignment. Then, the algorithm adds the p-vectors in $\mathscr{EF}$ and $\mathscr{F}$ and closed them under non-domination. When $\mathscr{X}$ is not empty, a variable is selected (line 3) and the algorithm sequentially attempts the assignment of its values to the p-functions in $\mathscr{F}$ (line 4,5). A lower bound set [9] of the cost of the current assignment is computed in the bounding evaluation function *LB* and compared with the current efficient frontier (line 6). If the current assignment may be extended, the search procedure proceeds by making a recursive call (line 7). Otherwise, the algorithm is in a dead-end and backtracks.

The performance of the search algorithm can be increased by reducing the explored search space. This reduction greatly depends on the *bounding evaluation function*. Therefore, the wisdom of the evaluation function to foresee a dead-end as soon as possible is a key factor in the branch and bound algorithm. MO-MBE can be executed inside branch and bound as a bounding evaluation function in order to provide lower bound sets of every subproblem. As MO-MBE is executed in each node, the control parameter z allows us to trade time for accuracy. In one hand, greater values of z will result in tighter lower bound sets. Therefore, the pruning capability of the algorithm will increase. However, the execution time will also increase. On the other, lower values of z will result in less tighter lower bound sets. However, the execution time will decrease and, as a consequence, reduce the time spent in every node.

5 Experimental Results

We have tested our approach in two different domains: *biobjective combinatorial auctions* and *biobjective weighted vertex cover* problems. The purpose of the experiments is to evaluate the performance of MO-BB using MO-MBE as an heuristic evaluation function (i.e. MO-BB$_{MOMBE}$) for solving MO-WCSP problems. To that end, we compare MO-BB$_{MOMBE}$ with the ε-constraint approach [7] based on search. Regarding MO-BB$_{MOMBE}$, experiments in the mono-objective case show that low values of the control parameter z usually provide reasonable good lower bounds with a very low cost [12]. Therefore, we follow the same criteria and set the control parameter $z = 2$ in all the experiments. For the ε-constraint approach, we use the well-known IlogSolver 6.1 as a solver engine. Moreover, the time spent for finding the ideal and nadir point that defines lower and upper bounds on the objective values of efficient solutions is not taken into account.

The time limit in all our experiments is 300 s. The execution time for unsolved instances is considered as that time limit. Therefore, for each domain, we report not just the cpu time, but also the percentage of solved instances within the time limit. We run all the experiments on a Pentium IV at 3GHz with 2*GB* of memory, running Linux.

5.1 Biobjective Combinatorial Auctions

Combinatorial auctions (CA) allow bidders to bid for indivisible subsets of goods [13]. In risk-conscious auctions, the auctioneer wants to control the risk of not being paid after a bid has been accepted, because it may cause large losses in revenue [14]. Consider a set of goods $\{1,2,\ldots,n\}$ that go on auction. There are m bids. Bid j is defined by the subset of requested goods $X_j \subseteq \{1,2,\ldots,n\}$, the money offer b_j and the probability of failure r_j. The auctioneer must decide which bids are to be accepted. If two bids have goods in common, only one of them can be accepted. The first objective is to maximize the auctioneer profit. The second objective is to minimize risk. Assuming independence, after a logarithmic transformation of probabilities, this objective can also be expressed as an additive function.

We have generated mono-objective CA using the PATH model of CATS generator [13] and randomly added payment failure probabilities to the bids in the range 0.0–0.3. We experiment on instances with 20 and 50 goods, varying the number of bids from 80 to 150. For each parameter configuration we generate samples of size 25.

Figure 4 reports the results obtained for instances with 20 and 50 goods corresponding to the plots on the right and on the left, respectively. MO-BB$_{MOMBE}$ outperformes ε-constraint in both configurations. For instances with 20 goods, MO-BB$_{MOMBE}$ solves all the instances within the time limit. However, ε-constraint only solves completely instances with 80 bids. Moreover, the solved percentage of ε-constraint decreases as the number of bids increases and it is quite low (20%)

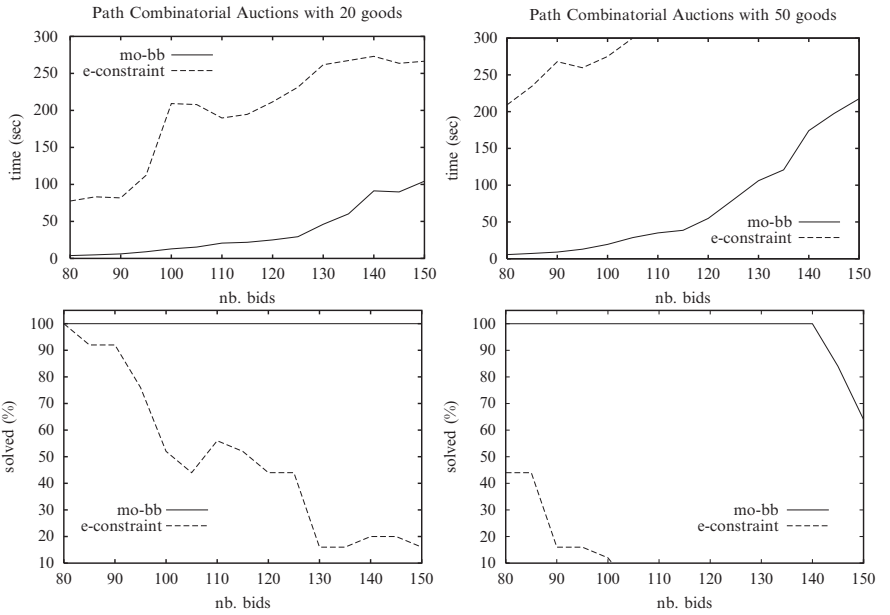


Fig. 4 Experimental results on bi-objective CA for 20 and 50 goods, respectively. Path distribution. Time limit 300 s

from 130 bids. It is important to note that, as the time for unsolved instances is set to 300 s, its effect in the mean cpu time is minimized. For instances with 50 goods, MO-BB$_{MOMBE}$ does not solve 4 instances with 145 bids and 9 with 150 bids. However, it is important to note that those instances can be solved in less than 400 s. ε-constraint does not solve completely any parameter configuration. Moreover, it fails in solving all instances from 105 bids.

5.2 Biobjective Weighted Vertex Cover

Given a graph $G = (V,E)$, a *vertex cover* is a subset of vertices $S \subseteq V$ such that $\forall (u,v) \in E$, either $u \in S$ or $v \in S$. The *minimum vertex cover* is a vertex cover of minimum size. In the *weighted* version every vertex u has an associated weight $w(u)$ and the *weighted minimum vertex cover* is a vertex cover S with minimum $F(S) = \sum_{u \in S} w(u)$. In the biobjective version each vertex u has two weights $w_1(u)$ and $w_2(u)$ and the task is to minimize the two associated objective functions. In our experiments we generated random graph instances with parameters (N, E, C) where N is the number of vertices, E is the number of edges and C is the maximum weight. Instances are generated by randomly selecting E edges. For each vertex, two costs are randomly generated from the interval $[0 \ldots C]$.

N (nb. vars)	E (nb. edges)	MO-BB$_{MOMBE}$ time (sec.)	%	ε-Constraint time (sec.)	%
60	95	0.92	100	155.82	80
70	95	1.68	100	289.75	8
80	95	3.17	100	300	0
90	95	6.72	100	288	4
60	250	1.92	100	28.22	100
70	250	4.56	100	221.94	40
80	250	9.23	100	280.46	8
90	250	20.04	100	300	0
60	500	2.03	100	2.56	100
70	500	5.87	100	26.63	100
80	500	17.12	100	216.21	52
90	500	42.35	100	300	0
60	950	1.51	100	0.27	100
70	950	3.87	100	2.65	100
80	950	10.49	100	16.09	100
90	950	32.46	100	122.51	100

Fig. 5 Experimental results on biobjective weighted minimum vertex cover problems. Parameter C is set to 4. Mean values on 25 instances for each parameter configuration. Time limit 300 s

We tested on samples of size 25 for the following parameter configurations $(\{60,70,80,90\},\{95,250,500,950\},4)$. Figure 5 reports the results obtained. The first and second column show the number of variables and edges, respectively. The third and fourth columns report the mean cpu time and the percentage of solved instances within the time limit using MO-BB$_{MOMBE}$. The fifth and sixth column report the same information for ε-constraint approach. The first thing to be observed is that MO-BB$_{MOMBE}$ is clearly superior for all parameter configurations. MO-BB$_{MOMBE}$ solves all instances within the time limit. However, ε-constraint is only able to solve completely instances with 950 edges. When we fix the number of constraints and increase the number of variables, the efficiency of both approaches decreases. When fixing the number of variables and increasing the number of constraints, the behaviour of both approaches differs. Regarding MO-BB$_{MOMBE}$, the solving time increases until instances with 500 edges. For instances with 950 edges the time diminishes. However, the solving time for ε-constraint always decreases.

6 Conclusions and Future Work

MultiObjective branch and bound (MO-BB) is a general search schema for multiobjective constraint optimization problems. The search space is represented as a tree. The algorithm searches depth-first the tree defined by the problem. Its output is the efficient frontier of the problem. The efficiency of the algorithm greatly depends on its pruning ability which, in turn, depends on the computation of a good lower bound set at each visited node.

MultiObjective mini-bucket elimination (MO-MBE) is an approximation algorithm for multiobjective constraint optimization problems. It has a control parameter

z which allow us to trade time and space for accuracy. Its output is a lower bound set of the efficient frontier of the problem. Therefore, it can be executed inside branch and bound as a bounding evaluation function in order to provide a lower bound set of every subproblem. We demonstrate the effectiveness of MO-BB using MO-MBE as a bounding evaluation function (i.e., MO-BB$_{MOMBE}$) in biobjective *combinatorial auctions* and *vertex cover* problems.

In our future work we want to evaluate the performance improvement of MO-BB$_{MOMBE}$ when using an initial good approximation of the efficient frontier. That initial approximation can be computed using approximate algorithms to compute upper bounds [7]. Moreover, we want to continue investigating the symbiosis between constraint programming and multiobjective optimization.

References

1. Bistarelli S, Fargier H, Montanari U, Rossi F, Schiex T, Verfaillie G (1999) Semiring-based CSPs and valued CSPs: Frameworks, properties and comparison. Constraints 4:199–240
2. Freuder E, Wallace R (1992) Partial constraint satisfaction. Artif Intell 58:21–70
3. Larrosa J, Schiex T (2003) In the quest of the best form of local consistency for weighted csp. In: Proceedings of the 18th IJCAI, Acapulco, Mexico
4. de Givry S, Heras F, Larrosa J, Zytnicki M (2005) Existential arc consistency: getting closer to full arc consistency in weighted csps. In: Proceedings of the 19th IJCAI, Edinburgh, UK
5. Dechter R, Rish I (2003) Mini-buckets: a general scheme for bounded inference. J ACM 50:107–153
6. Dechter R (2003) Constraint processing. Morgan Kaufmann, San Francisco
7. Ehrgott M, Gandibleux X (2002) Multiple criteria optimization. State of the art. Annotated bibliographic surveys. Kluwer, Dorderecht
8. Rollon E, Larrosa J (2006) Bucket elimination for multiobjective optimization problems. J Heurist 12:307–328
9. Ehrgott M, Gandibleux X (2001) Bounds and bound sets for biobjective combinatorial optimization problems. Lect Notes Econ Math Syst 507:241–253
10. Rossi F, van Beek P, Walsh T (2006) Handbook of constraint programming. Elsevier, Amsterdam
11. Fargier H, Lang J (1993) Uncertainty in constraint satisfaction problems: a probabilistic approach. Symbolic and Quantitative Approach to Reasoning and Uncertainity, European Conference ECSQARU 1993
12. Kask K, Dechter R (2001) A general scheme for automatic generation of search heuristics from specification dependencies. Artif Intell 129:91–131
13. Leuton-Brown K, Shoham MY (2000) Towards a universal test suite for combinatorial auction algorithms. ACM E-Comm:66–76
14. Holland A (2005) Risk management for combinatorial auctions. Dissertation, University College Cork, Ireland

Outer Branching: How to Optimize under Partial Orders?*

Ulrich Junker

Abstract Partial orders provide a convenient way to express preferences on multiple criteria. Prominent examples are Pareto-dominance and the preference relations of (T)CP-nets [1]. In advanced personalized recommender systems, the user may also specify a partial order over the possible values of a single criterion. We introduce a technique called outer branching to compute the non-dominated frontier of optimization problems with partial orders. It can be used to compute all Pareto-optimal solutions for n criteria by performing a systematic search over the criteria space. Dominance constraints avoid the generation of non-optimal solutions.

Keywords: Constraint programming · Pareto optimization · Preference handling

1 Introduction

Partial orders provide a convenient way to express preferences on multiple criteria. For example, Pareto-optimality uses a partial order to compare tuples of criteria values. Partial orders may also be used to compare the possible values of a single criterion. This happens if customer preferences are not completely specified (such as the customer prefers a red to a blue car, but does not specify a preference concerning a green car). This also impacts notions such as lexicographic optimality, which imposes a partial order on the criteria space, but not a total one, if the individual criteria have partial orders.

In this paper, we consider combinatorial problems with partial orders and we investigate methods to explore the whole space of optimal solutions. We encounter

U. Junker
ILOG 1681, route des Dolines, F-06560 Valbonne, France
e-mail: ujunker@ilog.fr

*This work has partially been supported by the French National Research Agency (Agence Nationale de la Recherche) under contract no NT05-4_41833.

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

this as a subproblem in product configuration when establishing a catalog of component configurations that are Pareto-optimal w.r.t. price, quality, power consumption [8]. There are two fundamental approaches in order to solve optimization problems with partial orders. The first approach consists of a generalized Branch-and-Bound (B&B) search which maintains a frontier of the best non-dominated values found so far. In Constraint Programming (CP), the frontier prunes the dominated region by bound propagation [6]. The other approach breaks the optimization problem with partial orders into multiple optimization problems with total orders. These subproblems can usually be solved by standard optimizers and this second approach can therefore be implemented with less effort as the multi-objective B&B. In multi-objective programming, the second approach is pursued by scalarization methods [12] and by the ε-constraint method [2]. Both do a search over an auxiliary space such as a weight space or a parameter space in order to produce the whole frontier. However, not every weight and parameter combination leads to a new optimal solution.

We propose an alternative way of computing the non-dominated set by a sequence of standard optimization problems, which satisfies three properties:

(a) *No Failure:* each step, except for the last one, computes a solution.
(b) *No Dominance:* no dominated solution is computed.
(c) *No Redundancy:* a solution is not computed twice.

The approach adds 'outer' constraints to the set of non-dominated solutions and is able to split it into disjoint subsets. If chosen carefully, these outer constraints can be transformed into inner constraints which can be taken into account by a standard optimizer.

We first introduce combinatorial problems with preferences (Sect. 2) and review existing optimization approaches (Sect. 3). We then elaborate the outer branching principle for general partial orders (Sect. 4) and specialize it for computing the Pareto-optimal frontier for n criteria (Sect. 5).

2 Combinatorial Optimization Under Partial Orders

We consider decision-making problems with combinatorial decision spaces, combinatorial criteria spaces, and incomplete (i.e. partial) preference orders.

Let $D_1,\ldots,D_m$ be m finite domains and let $\mathscr{D}$ be their Cartesian product $D_1 \times \ldots \times D_m$. This combinatorial space is subject of constraints C which we express by logical conditions $\phi(\mathbf{x})$ on the variables $\mathbf{x} := (x_1,\ldots,x_m)$. These may be linear constraints, symbolic constraints, and boolean combinations of those constraints as offered by modelling languages such as OPL and by Constraint Programming tools. A tuple $\mathbf{v} := (v_1,\ldots,v_m)$ from $\mathscr{D}$ is a *solution* of the constraint $\phi(\mathbf{x})$ iff the condition $\phi(\mathbf{v})$, which results from replacing the variables by the values, is true. The *decision space* is the set of solutions of C.

The decisions are evaluated in terms of multiple criteria $z_1, \ldots, z_n$. Let $\Omega_1, \ldots, \Omega_n$ be n outcome domains and let Ω be their Cartesian product $\Omega_1 \times \cdots \times \Omega_n$, which constitutes the *outcome space*. Each criterion z_i is a function from the decision space to the outcome domain Ω_i and thus maps a decision $\mathbf{v}$ to an outcome value $z_i(\mathbf{v})$. Examples for those functions are (piecewise) linear expressions, minimum and maximum of variables, table expressions, and their combinations. The *criteria space* is the set of outcome vectors $z(\mathbf{v}) := (z_1(\mathbf{v}), \ldots, z_n(\mathbf{v}))$ that are obtained for all elements $\mathbf{v}$ of the decision space.

The decision maker can compare the values in an outcome domain Ω_i and formulate preferences between them, which we model by a binary relation P on the outcome domain Ω_i. If (ω_1, ω_2) is in P, then the decision maker prefers ω_1 as least as much as ω_2. The transitive and reflexive closure of P is a (partial) preorder $\succsim$, i.e. a transitive and reflexive binary relation. We are mainly interest in the strict part $\succ$ of this preorder, namely the set of all pairs (ω_1, ω_2) in $\Omega_i \times \Omega_i$ such that $\omega_1 \succsim \omega_2$ holds, but not $\omega_2 \succsim \omega_1$. The absence of a strict preference between two outcomes can either signify indifference or incompleteness. The strict part of a preorder is a strict partial order, i.e. an irreflexive and transitive relation. We write $\omega_1 \succeq \omega_2$ as a short-hand for $\omega_1 \succ \omega_2$ or $\omega_1 = \omega_2$. The relation $\succeq$ is a partial order, i.e. an antisymmetric preorder. It is a subset of the preorder $\succsim$, but the inverse does not hold in general. If all pairs ω_1, ω_2 satisfy one of $\omega_1 \succ \omega_2$ or $\omega_2 \succ \omega_1$ or $\omega_1 = \omega_2$, then $\succ$ is a strict total order (and $\succeq$ is a total order).

A simple example is the choice of a wind-surfing destination among CA (California), FL (Florida), HI (Hawaii), TX (Texas). The decision space consists of these four alternatives. The user directly compares these decisions, meaning that the outcome space is equal to the decision space and that the identity function *id* is the sole criterion. The user prefers Hawaii to Florida and to California, which is preferred to Texas. This leads to the strict partial order $\succ_w$ satisfying HI $\succ_w$ FL, HI $\succ_w$ CA, CA $\succ_w$ TX, and HI $\succ_w$ TX.

In general, the decision maker will formulate preferences on multiple criteria $z_1, \ldots, z_n$. We thus obtain a strict partial order $\succ_i$ on Ω_i for each criterion z_i. There are different ways to combine those orders on Ω_i into an order on the Cartesian outcome space $\Omega_1 \times \cdots \times \Omega_n$. Outcome vectors can be compared by different optimality notions such as lexicographic optimality, Pareto-optimality, or leximin-optimality [3, 4]. A vector $(\omega_1^*, \ldots, \omega_n^*)$ is lexicographically better than a vector $(\omega_1, \ldots, \omega_n)$ iff there exists an i such that $\omega_i^* \succ_i \omega_i$ and $\omega_j^* = \omega_j$ for $j = 1, \ldots, i-1$. A vector $(\omega_1^*, \ldots, \omega_n^*)$ dominates a vector $(\omega_1, \ldots, \omega_n)$ in the Pareto-sense iff the two vectors are different (i.e. $\omega_i^* \neq \omega_i$ for an i) and all outcomes in the first vector are at least as good as the outcomes in the second vector (i.e. $\omega_i^* \succeq_i \omega_i$ for all i). We thus obtain strict partial orders such as $\succ_{lex}$ and $\succ_{pareto}$ on the criteria space Ω. It is important to note that the lexicographical order $\succ_{lex}$ is a strict total order iff all the criterion orders $\succ_i$ are strict total orders, but that the Pareto-order cannot be a strict total order if there are at least two criteria.

Given a strict partial order $\succ$ on the outcome space Ω, a solution $\mathbf{v}$ in the decision space dominates another solution $\mathbf{w}$ in the decision space iff $z(\mathbf{v}) \succ z(\mathbf{w})$ holds. We consider the set of *non-dominated solutions* and the set of their outcomes, which

we call the *optimal outcomes*. We characterize the non-dominated solutions by a logical optimization operator $Max_{z,\succ}$ that maps the constraint C to a new constraint. The solutions of $Max_{z,\succ}(C)$ are the non-dominated solutions of C. We now assume that the constraint language is rich enough to express constraints using the partial order $\succ$ and the expression $z(\mathbf{x})$. A vector $\mathbf{v} \in \mathscr{D}$ is a solution of $Max_{z,\succ}(C)$ iff C has the form $\phi(\mathbf{x})$ and $\phi(\mathbf{v}) \wedge \neg\exists \mathbf{w} \in \mathscr{D} : \phi(\mathbf{w}) \wedge z(\mathbf{w}) \succ z(\mathbf{v})$ is true. As this constraint contains a quantifier, it cannot be expressed in usual CP and MIP solvers. However, it provides an elegant logical characterization of the problem to solve, i.e. the computation of the set Ω^* of optimal outcomes, as the following property holds:

$$Max_{z,\succ}(C) \equiv C \wedge \bigvee_{\omega \in \Omega^*} z(\mathbf{x}) = \omega \tag{1}$$

If the order $\succ$ is a strict total order $>$, then the set of optimal outcomes is a singleton and we refer to its element as the outcome of $Max_{z,>}(C)$. Our optimization operator is similar to those used in [3], but is fully embedded in the constraint logic and can thus be combined with other constraints.

3 Existing Optimization Approaches

The Branch-and-Bound method (B&B) successively splits the decision space into disjoint subspaces. A branching step transforms a problem $Max_{z,\succ}(C)$ into the disjunction $Max_{z,\succ}(C \wedge \delta) \vee Max_{z,\succ}(C \wedge \neg\delta)$ by adding the branching constraint δ to one branch and its negation to the other branch. The disjunction is logically implied by the original problem, but is *not* equivalent to it as a disjunct can have solutions that are dominated by the solutions of the other disjunct. For instance, consider a criterion $d : \mathbb{Z} \to \mathbb{Z}$ such that $d(x) := 2 \times x$ and the problem $Max_{d,>}(x \leq 50)$ of finding the greatest outcome $d(x)$ while keeping x smaller than 50. We use the branching constraint $x \geq 30$. An outcome $d(x)$ of 2 times 50 is obtained for all solutions x of the left subproblem $Max_{d,>}(x \leq 50 \wedge x \geq 30)$. The right subproblem $Max_{d,>}(x \leq 50 \wedge x < 30)$ has solutions with outcome $d(x)$ of 2 times 29, which do not belong to the set of non-dominated solutions of C. In general, $Max_{z,\succ}(C \wedge \delta)$ does not imply $Max_{z,\succ}(C)$ as δ can remove non-dominated solutions from the decision space of C. Classic B&B is used for optimization problems with total orders. It maintains the outcome $z(\mathbf{v})$ of the best solution found so far and uses it to prune subproblems that do not contain any better solution. When used with partial orders, B&B needs to keep a whole frontier of currently best outcomes [6] and requires sophisticated operations for updating the frontier.

Another method explores different extensions of the partial order and transforms the problem into multiple optimization problems under total orders. A *linear extension* of a (strict) partial order $\succ$ is a (strict) total order $>$ that is a superset of the (strict) partial order. For example, the order $>_w$ defined as HI $>_w$ CA $>_w$ TX $>_w$ FL is a linear extension of the wind-surfing preferences $\succ_w$. We then solve the problem $Max_{z,>}(C)$ by a standard optimizer based on classic B&B. The resulting

solutions are non-dominated solutions of C. Moreover, each solution of C that is non-dominated under the strict partial order $\succ$ is also non-dominated under some linear extension. As the space of linear extensions is factorial in the number d of outcomes in Ω, many linear extensions will produce the same solution.

This number can be reduced by generalizing the ε-constraint method [2], which tests whether an outcome $\varepsilon \in \Omega$ is optimal by adding the lower bound constraint $z(\mathbf{x}) \succeq \varepsilon$ to C. The method uses a linear extension $>$ of the strict partial order $\succ$ and determines the outcome ω of $Max_{z,>}(C \wedge z(\mathbf{x}) \succeq \varepsilon)$. It thus considers only solutions $\mathbf{v}$ in the decision space that have an outcome $z(\mathbf{v})$ that is equal to or better than ε w.r.t. the partial order and it compares those solutions w.r.t. the total order. If the best outcome ω is equal to ε, then ε has succeeded the test since the following equivalence holds for lower bounds on $z(\mathbf{x})$:

$$Max_{z,\succ}(C \wedge z(\mathbf{x}) \succeq \varepsilon) \equiv Max_{z,\succ}(C) \wedge z(\mathbf{x}) \succeq \varepsilon \tag{2}$$

Hence, ε is the outcome of a non-dominated solution in the decision space of C iff it is the outcome of a non-dominated solution in the subset of the decision space, which contains all those solutions of C having an outcome that is equal to or better than ε w.r.t. the partial order. This method performs only d optimization step, namely one for each outcome in Ω. However only some of those optimizations will produce a non-dominated solution.

4 The Outer Branching Principle

As an alternative, we introduce an outer branching scheme, which splits the set of non-dominated solutions into disjoint subsets. When using the branching constraint δ, we consider the subset of non-dominated solutions that satisfies δ in the left branch and the subset of non-dominated solutions that violate δ in the right branch. Formally, we transform the original problem $Max_{z,\succ}(C)$ into the logically equivalent disjunction $(Max_{z,\succ}(C) \wedge \delta) \vee (Max_{z,\succ}(C) \wedge \neg\delta)$. We apply outer branching to the example $Max_{d,>}(x \leq 50)$ and the constraint $x \geq 30$. The left subproblem $Max_{d,>}(x \leq 50) \wedge x \geq 30$ has the outcome of 2 times 50 and the right subproblem $Max_{d,>}(x \leq 50) \wedge x < 30$ has no solution.

Outer branching leads to conjunctions of optimization problems and of 'outer' constraints. It is only useful if the outer constraints simplify the original optimization problems. This simplification is possible if the branching constraint has the form $z(\mathbf{x}) = \omega^*$ and ω^* is an optimal outcome of the original problem. As explained above, we obtain such a start solution by applying a standard optimizer to a linear extension $>$ of the strict partial order $\succ$. The resulting outcome ω^* permits us to apply the outer branching scheme:

$$Max_{z,\succ}(C) \equiv [Max_{z,\succ}(C) \wedge z(\mathbf{x}) = \omega^*] \vee [Max_{z,\succ}(C) \wedge z(\mathbf{x}) \neq \omega^*] \tag{3}$$

The first branch is satisfied by exactly those non-dominated solutions that have the outcome ω^* and is thus reduced as follows:

$$[Max_{z,\succ}(C) \wedge z(\mathbf{x}) = \omega^*] \equiv [C \wedge z(\mathbf{x}) = \omega^*] \tag{4}$$

In our example, we suppose that Hawaii is not possible as wind-surfing destination. We obtain California as outcome of the problem $Max_{id,>_w}(C_w)$ where C_w is $x \neq \text{HI}$. Hence, the left branch reduces to $C_w \wedge id(x) = \text{CA}$.

In the second branch, we obtain all those non-dominated solutions that do not have the outcome ω^*. None of these solutions has an outcome that is equal to or dominated by ω^*. We can therefore 'internalize' the outer constraint $z(\mathbf{x}) \neq \omega^*$ by the constraint $z(\mathbf{x}) \not\preceq \omega^*$ supposing that our constraint language permits such a formulation ($\omega_1 \prec \omega_2$ stands for $\omega_2 \succ \omega_1$):

$$Max_{z,\succ}(C) \wedge z(\mathbf{x}) \neq \omega^* \equiv Max_{z,\succ}(C \wedge z(\mathbf{x}) \not\preceq \omega^*) \tag{5}$$

In the example, the right branch reduces to the problem $Max_{id,\succ_w}(C_w \wedge id(x) \not\preceq_w \text{CA})$. We then apply the whole procedure to this new problem. In this case, neither California, nor Texas are possible values for x since they both satisfy $id(x) \preceq_w \text{CA}$. Hence, Florida is the optimal wind-surfing destination of this tightened problem and we again do an outer branching based on this outcome. The left branch imposes Florida, whereas the right branch eliminates this last possible value, thus leading to an inconsistent problem. As a consequence, the procedure stops.

An option is to record the fact that ω^* is the best value for z w.r.t the total order $>$:

$$Max_{z,\succ}(C) \wedge z(\mathbf{x}) \neq \omega^* \equiv Max_{z,\succ}(C \wedge z(\mathbf{x}) \not\preceq \omega^* \wedge z(\mathbf{x}) \leq \omega^*) \tag{6}$$

We can remove the equality from $z(\mathbf{x}) \leq \omega^*$ thanks to the constraint $z \not\preceq \omega^*$:

$$Max_{z,\succ}(C) \wedge z(\mathbf{x}) \neq \omega^* \equiv Max_{z,\succ}(C \wedge z(\mathbf{x}) \not\preceq \omega^* \wedge z(\mathbf{x}) < \omega^*) \tag{7}$$

The complete algorithm is given in Fig. 1. We consider two variants, which both determine the outcomes of the non-dominated solutions. Algorithm OB1 chooses a different linear extension in each iteration and is more flexible. Algorithm OB2

Algorithm OB1$(z, \succ, C)$

1. $D :=$ True; $U :=$ True; $\Omega^* := \emptyset$;
2. **repeat**
3. let $>$ be a linear extension of $\succ$;
4. solve $S := Max_{z,>}(C \wedge D \wedge U)$;
5. **if** S has no solution
6. **then** return Ω^*;
7. let ω^* be the outcome of S;
8. $\Omega^* := \Omega^* \cup \{\omega^*\}$;
9. $D := D \wedge z(\mathbf{x}) \not\preceq \omega^*$;
10. $U := U \wedge z(\mathbf{x}) < \omega^*$;

Algorithm OB2$(z, \succ, C)$

1. $D :=$ True; $U :=$ True; $\Omega^* := \emptyset$;
2. let $>$ be a linear extension of $\succ$;
3. **repeat**
4. solve $S := Max_{z,>}(C \wedge D \wedge U)$;
5. **if** S has no solution
6. **then** return Ω^*;
7. let ω^* be the outcome of S;
8. $\Omega^* := \Omega^* \cup \{\omega^*\}$;
9. $D := D \wedge z(\mathbf{x}) \not\preceq \omega^*$;
10. $U := z(\mathbf{x}) < \omega^*$;

Fig. 1 Computing the optimal outcomes

uses the same linear extension in all steps and enumerates the non-dominated solutions in decreasing order w.r.t. the chosen linear extension. Both versions maintain a conjunction D of dominance constraints $z(\mathbf{x}) \not\preceq \omega^*$ for all the optimal outcomes ω^* found so far. They also maintain a conjunction of upper-bound constraints of the form $z(\mathbf{x}) < \omega^*$ which are optional, but which may help to reduce search effort. As OB1 uses different linear extensions, it cannot simplify two upper-bound constraints $z(\mathbf{x}) <_1 \omega_1^*$ and $z(\mathbf{x}) <_2 \omega_2^*$. However, OB2 only needs to keep the last upper-bound constraint $z(\mathbf{x}) < \omega^*$ as all the other upper-bound constraints $z(\mathbf{x}) < \omega$ use an outcome ω that is better than ω^* w.r.t. the total order $>$.

The algorithms meet the requirements listed in the introduction and ensure that each step finds a new non-dominated solution. As the conjunction of dominance constraints is growing in each iteration, we will investigate efficient representations for the important case of Pareto-optimality.

5 Outer Branching for Pareto-Optimality

In this section, we show how to determine the complete set of Pareto-optimal solutions for n criteria. For convenience, we suppose that each criterion z_i has a total order $>_i$, although the approach handles partial orders as well.

The lexicographic ordering $>_{lex}$ is a linear extension for Pareto-dominance $\succ_{pareto}$ and we exploit this relationship to apply the outer branching scheme (3) and the algorithm OB2. The constraints $z(\mathbf{x}) \not\preceq_{pareto} \omega$ and $z(\mathbf{x}) <_{lex} \omega$ can be expressed as logical constraints on the criteria vector $z(\mathbf{x}) := (z_1(\mathbf{x}), \ldots, z_n(\mathbf{x}))$ and the value vector $\omega := (\omega_1, \ldots, \omega_n)$:

$$\begin{aligned} z(\mathbf{x}) \not\preceq_{pareto} \omega &\equiv \bigvee_{j=1}^{n} z_j(\mathbf{x}) >_j \omega_j \\ z(\mathbf{x}) <_{lex} \omega &\equiv \bigvee_{i=1}^{n} \Big(\bigwedge_{j=1}^{i-1} z_j(\mathbf{x}) = \omega_j \wedge z_i(\mathbf{x}) <_i \omega_i \Big) \end{aligned} \tag{8}$$

As problem solving is progressing, we obtain a sequence of solutions $\omega^{(1)}, \ldots, \omega^{(k)}$ in the k-th iteration of algorithm OB2. As discussed in the last section, we keep only the lexicographical constraint $z(\mathbf{x}) <_{lex} \omega^{(k)}$ generated for the k-th Pareto-optimal solution, but we accumulate the dominance-constraints, thus obtaining the conjunction $\bigwedge_{i=1}^{k} z(\mathbf{x}) \not\preceq_{pareto} \omega^{(i)}$. As the lexicographical constraint implies $z_1(\mathbf{x}) \leq \omega_1^{(i)}$, we remove the constraints $z_1(\mathbf{x}) >_1 \omega_1^{(i)}$ from the dominance constraints. Similarly, the last disjunct of the lexicographical constraint can be removed as it does not improve any of the criteria as required by the dominance constraint:

$$\begin{aligned} & z(\mathbf{x}) <_{lex} \omega^{(k)} \wedge \bigwedge_{i=1}^{k} z(\mathbf{x}) \not\preceq_{pareto} \omega^{(i)} \\ \equiv & \bigvee_{i=1}^{n-1} \Big(\bigwedge_{j=1}^{i-1} z_j(\mathbf{x}) = \omega_j^{(k)} \wedge z_i(\mathbf{x}) <_i \omega_i^{(k)} \Big) \wedge \bigwedge_{i=1}^{k} \bigvee_{j=2}^{n} z_j(\mathbf{x}) >_j \omega_j^{(i)} \end{aligned} \tag{9}$$

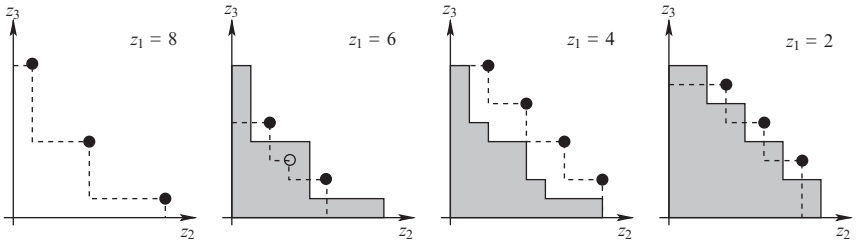


Fig. 2 Pareto-optimal solutions

We consider a simple example with three criteria z_1, z_2, z_3, all of which have the maximization order $>$. The Pareto-frontier is described in form of four slices in Fig. 2. We choose the lexicographical ordering based on the importance ranking z_1, z_2, z_3, which produces the optimal solution $\omega^{(1)} := (8,8,1)$. In the right branch, we add the constraint

$$(z_1(\mathbf{x}) < 8 \vee (z_1(\mathbf{x}) = 8 \wedge z_2(\mathbf{x}) < 8)) \wedge (z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 1)$$

The lexicographical solution of the original problem and this new constraint is $\omega^{(2)} := (8,4,4)$. We now obtain the following constraint in the right branch:

$$\begin{gathered}(z_1(\mathbf{x}) < 8 \vee (z_1(\mathbf{x}) = 8 \wedge z_2(\mathbf{x}) < 4)) \wedge \\ (z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 1) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 4)\end{gathered}$$

The next solution found by the procedure is $\omega^{(3)} := (8,1,8)$ leading to the following constraint:

$$\begin{gathered}(z_1(\mathbf{x}) < 8 \vee (z_1(\mathbf{x}) = 8 \wedge z_2(\mathbf{x}) < 1)) \wedge \\ (z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 1) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 4) \wedge (z_2(\mathbf{x}) > 1 \vee z_3(\mathbf{x}) > 8)\end{gathered}$$

We have thus determined the three Pareto-optimal solutions for $z_1(\mathbf{x}) = 8$. Two further iterations add the two Pareto-optimal solutions that satisfy $z_1(\mathbf{x}) = 6$. Please note that $(6,3,3)$ is not a Pareto-optimal solution. It is not returned by the outer branching algorithm since it violates $z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 4$. The shadowed region in Fig. 2 shows the area that is pruned by the dominance constraints. The fifth iteration produces

$$\begin{gathered}(z_1(\mathbf{x}) < 6 \vee (z_1(\mathbf{x}) = 6 \wedge z_2(\mathbf{x}) < 2)) \wedge \\ (z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 1) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 4) \wedge (z_2(\mathbf{x}) > 1 \vee z_3(\mathbf{x}) > 8) \wedge \\ (z_2(\mathbf{x}) > 5 \vee z_3(\mathbf{x}) > 2) \wedge (z_2(\mathbf{x}) > 2 \vee z_3(\mathbf{x}) > 5)\end{gathered}$$

Four further iterations yield the four Pareto-optimal solutions under $z_1(\mathbf{x}) = 4$. The resulting dominance and lex-constraint is:

$$(z_1(\mathbf{x}) < 4 \vee (z_1(\mathbf{x}) = 4 \wedge z_2(\mathbf{x}) < 2)) \wedge$$
$$(z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 1) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 4) \wedge (z_2(\mathbf{x}) > 1 \vee z_3(\mathbf{x}) > 8) \wedge$$
$$(z_2(\mathbf{x}) > 5 \vee z_3(\mathbf{x}) > 2) \wedge (z_2(\mathbf{x}) > 2 \vee z_3(\mathbf{x}) > 5) \wedge$$
$$(z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 2) \wedge (z_2(\mathbf{x}) > 6 \vee z_3(\mathbf{x}) > 4) \wedge$$
$$(z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 6) \wedge (z_2(\mathbf{x}) > 2 \vee z_3(\mathbf{x}) > 8)$$

As the first criterion $z_1(\mathbf{x})$ is omitted in the dominance constraint, we encounter a dominance effect inside the dominance constraint. Indeed, the four new solutions projected to z_2, z_3 dominate the previous solutions projected to z_2, z_3. As a consequence, we can simplify the constraint as follows:

$$(z_1(\mathbf{x}) < 4 \vee (z_1(\mathbf{x}) = 4 \wedge z_2(\mathbf{x}) < 2)) \wedge$$
$$(z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 2) \wedge (z_2(\mathbf{x}) > 6 \vee z_3(\mathbf{x}) > 4) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 6) \wedge$$
$$(z_2(\mathbf{x}) > 2 \vee z_3(\mathbf{x}) > 8)$$

Three further iterations add the three Pareto-optimal solutions under $z_1(\mathbf{x}) = 2$ and result into the following constraint:

$$(z_1(\mathbf{x}) < 2 \vee (z_1(\mathbf{x}) = 2 \wedge z_2(\mathbf{x}) < 3)) \wedge$$
$$(z_2(\mathbf{x}) > 8 \vee z_3(\mathbf{x}) > 2) \wedge (z_2(\mathbf{x}) > 6 \vee z_3(\mathbf{x}) > 4) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 6) \wedge$$
$$(z_2(\mathbf{x}) > 2 \vee z_3(\mathbf{x}) > 8) \wedge$$
$$(z_2(\mathbf{x}) > 7 \vee z_3(\mathbf{x}) > 3) \wedge (z_2(\mathbf{x}) > 4 \vee z_3(\mathbf{x}) > 4) \wedge (z_2(\mathbf{x}) > 3 \vee z_3(\mathbf{x}) > 7)$$

The final problem has no solution and the procedure stops. Our approach performs k lexicographic optimizations to find k Pareto-optimal solutions, but increases the number of disjunctive constraints in each step. As each lexicographic optimization needs $O(n)$ calls of a standard optimizer, the algorithm needs $O(k \cdot n)$ calls of a standard optimizer to find k Pareto-optimal solutions.

6 Related Work

Outer branching uses a dominance constraint similar to Gavenelli's approach [6]. Whereas Gavanelli's approach performs a single search based on multiple objectives, outer branching performs multiple searches based on a single objective, which permits a more informed search and which is essential for applying modern MIP solvers.

Although the disjunctive dominance constraint can be encoded by linear constraints, the resulting problem does not necessarily have a good linear relaxation. Other approaches therefore seek to eliminate the dominance constraint, but lead to subproblems that produce redundant solutions, dominated solutions, or no solution. In [10], the ε-constraint method is successively applied to all vectors ω obtained by combining criteria values from previous solutions. This leads to (k^{n-1}) standard optimizer calls in the worst-case and can lead to dominated solutions and failing subproblems. Preference-based search in [7] uses a more complex form of outer

branching with branching constraints of the form $z_i \geq \omega_i$. However, the negation $z_i < \omega_i$ cannot be moved inside the optimization operator and the method may encounter a significant number of subproblems without solution as illustrated in [7].

Hence, if no redundancy, no dominated solutions, and no failing subproblems are desired, then outer branching provides a way for meeting these desiderata. The method can be improved by factorizing effort between the subproblems and by dynamically simplifying the dominance constraint.

7 Conclusion

We introduced the outer branching method for combinatorial optimization problems with partial orders and used it to compute a multi-dimensional Pareto-frontier in a systematic way, thus addressing an important research topic [5–7, 10]. Dominance constraints avoid that the same solution is computed twice and ensure that only $O(k \cdot n)$ calls of a standard optimizer are needed to compute k Pareto-optimal solutions. The method generalizes efficient versions of the ε-constraint method [11, 13], which were limited to the bicriterion case. This study gives also insights about the difficulties that need to be addressed when applying the approach to other optimality notions such as (T)CP-nets [1] or dynamic lexicographical orderings [9].

References

1. Boutilier C, Brafman R, Domshlak C, Hoos H, Poole D (2004) Preference-based constrained optimization with CP-Nets. Comput Intell 20:137–157
2. Chankong V, Haimes Y (1983) Multiobjective decision making theory and methodology. Elsevier, New York
3. Ehrgott M (1997) A characterization of lexicographic max-ordering solutions. In: Methods of multicriteria decision theory. Proceedings of the 6th Workshop of the DGOR Working-Group Multicriteria Optimization and Decision Theory, Häsel-Hohenhausen, Egelsbach, pp 193–202
4. Ehrgott M, Gandibleux X (2000) A survey and annoted bibliography of multiobjective combinatorial optimization. OR Spektrum 22:425–460
5. Ehrgott M, Ruzika S (2005) An improved epsilon-constraint method for multiobjective programming. Technical report, University of Kaiserslautern, Germany
6. Gavanelli M (2002) An algorithm for multi-criteria optimization in CSPs. In: Proceedings of ECAI'02, IOS , Lyon, France, pp 136–140
7. Junker U (2004) Preference-based search and multi-criteria optimization. Ann Oper Res 130:75–115
8. Junker U (2005) Preference-based inconsistency proving: When the failure of the best is sufficient. In: Proceedings of IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling, pp 106–111
9. Junker U, Mailharro D (2003) Preference programming: advanced problem solving for configuration. AI-EDAM 17:13–29
10. Laumanns M, Thiele L, Zitzler E (2005) An adaptive scheme to generate the pareto front based on the epsilon-constraint method. In: Branke J, Deb K, Miettinen K, Steuer R (eds) Practical approaches to multi-objective optimization. Dagstuhl Seminar Proceedings 04461

11. Srinivasan V, Thompson G (1976) Algorithms for minimizing total cost, bottleneck time, and bottleneck shipment in transportation problems. Naval Res Logist Quart 23:567–598
12. Steuer R (1986) Multi criteria optimization: theory, computation, and application. Wiley, New York
13. Van Wassenhove L, Gelders L (1980) Solving a bicriterion scheduling problem. Eur J Oper Res 4(1):42–48

On Utilizing Infeasibility in Multiobjective Evolutionary Algorithms

Thomas Hanne

Abstract In this article, we consider the problem of infeasible solutions (i.e. solutions which violate one or several restrictions of an optimization problem) which can hardly be avoided when new solutions are generated by stochastic and other means during the run of an optimization algorithm. Since typical approaches for dealing with infeasibility such as using a repair mechanism, a punishment approach, or a simple recalculation of solutions are not fully satisfying in many problems, we suggest a new approach of tolerating and actively using infeasible solutions within the framework of multiobjective evolutionary algorithms.

The novel evolutionary algorithm allows solving a multiobjective optimization problem (MOP) with continuous variables by approximating the efficient set. The algorithm uses populations of variable size and new rules for selecting solutions for the subsequent generations. In particular, some of the selected solutions may be infeasible such that the Pareto front is approached at the same time from two "sides", the feasible set and a subset of the infeasible set. Since the considered infeasible solutions correspond to a dual optimization problem, we call the new algorithm primal–dual multiobjective optimization algorithm, or PDMOEA. The algorithm is demonstrated by considering a numerical test problem and is compared with two other approaches for dealing with infeasibility. The example shows a specific strength of the new approach: By tunneling through infeasible regions, the population may more easily extent to new separated parts of the Pareto set.

Keywords: Constrained optimization · Efficient set · Evolutionary algorithms · Feasibility · Multiobjective optimization

T. Hanne
University of Applied Sciences Northwestern Switzerland, Department of Information Systems, Riggenbachstrasse 16 – 4600 Olten, Switzerland
e-mail: thomas.hanne@fhnw.ch

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

1 Introduction

In optimization algorithms, the generation of new solutions by stochastic and other means is often a significant problem because of violating restrictions. It is often not clear how to obtain new feasible solutions in a straight-forward manner. In many cases, it is suggested to use repair mechanisms which calculate a feasible solution from an infeasible one (see, e.g., [10] and [9]). In many cases this approach is, however, time consuming. In some situations, it is not clear at all how to repair an infeasible solution in a canonical way. Other means to cope with infeasible solutions are, for instance, the anew generation of a solution (see, e.g., [4]) or the usage of a penalty function (see, e.g., [12]) which may provide similar disadvantages. Another approach for dealing with constraints in MOEAs is discussed in [2]. In complex real-life multiobjective optimization problems dealing with many soft and hard constraints (e.g. in the areas of production planning, scheduling, or vehicle routing) the issue of infeasibility is of paramount importance for a successful application.

In this article, we present a novel evolutionary algorithm for approximating the efficient set of a multiobjective optimization problem (MOP) with continuous variables. The algorithm is based on populations of variable size and exploits new rules for selecting alternatives generated by mutation and recombination. The specific approach for dealing with infeasibility is that the algorithm solves at the same time the original problem and a dual problem such that solutions advance the efficient border from two "sides", the feasible set and a subset of the infeasible set. Together with additional assumptions on the considered MOP and further specifications on the algorithm, theoretical results on the approximation quality and the convergence of both subpopulations, the feasible and the infeasible one, can be analyzed [7].

The article is organized as follows: In Sect. 2, the MOP and notations related to solution sets are introduced. A dual MOP is analyzed in Sect. 3. Section 4 presents a description of the new MOEA. In Sect. 5 some numerical test results are shown. Section 6 gives the conclusions.

2 Some Notations

Mathematically, multiobjective optimization problems (MOPs) can be described by

$$\text{“}min\text{”} f(x), \text{s.t.} x \in A \tag{1}$$

where $f : R^n \to R^q$ is a vector-valued objective function and "*min*" means that each of the objective functions (the components of f) should be minimized. In a similar way, maximization problems can be defined. $A \subseteq R^n$ is called the feasible set which is usually defined by restriction functions, i.e. $A = \{x \in R^n : g_j(x) \leq 0, j \in \{1,...,m\}\}$. Note that, we assume f being defined on the whole domain R^n such that the objective function can be evaluated for infeasible alternatives.

Usually for an MOP, there does not exist a unique solution which optimizes all objective functions at the same time. Therefore, mostly the set of efficient or Pareto-optimal alternatives is regarded as the solution set of the problem (1). For specifying this and related sets, the Pareto relation " $\leq$ " defined by

$$x \leq y :<=> x_i \leq y_i \; \forall i \in \{1,...,q\} \text{ and } x_i < y_i \exists i \in \{1,...,q\} \tag{2}$$

for $x, y \in R^q$ is used. The component-wise generalization of the scalar " $\leq$ " is defined by

$$x \stackrel{\leq}{=} y :<=> x_i \leq y_i \; \forall i \in \{1,...,q\}. \tag{3}$$

Using the Pareto relation, the efficient set is defined by

$$E(A,f) := \{x \in A : \nexists y \in A : f(y) \leq f(x)\}. \tag{4}$$

The image of $E(A,f)$ in objective space R^q is also denoted as Pareto set or Pareto front. See [3] for a survey on efficient sets and other mathematical solution concepts for MOPs. The set of dominating alternatives with respect to a given set $B \subseteq R^n$ is defined as

$$Dom(B,f) := \{x \in R^n : \exists y \in B : f(x) \leq f(y) \wedge \nexists z \in B : f(z) \stackrel{\leq}{=} f(x)\}. \tag{5}$$

Basically, the dominating set consists of all nonfeasible vectors being (strictly) better than solutions from the efficient set. The nondominated set with respect to a given set B is denoted as

$$Nondom(B,f) := \{x \in R^n : \nexists y \in B : f(y) \stackrel{\leq}{=} f(x)\}. \tag{6}$$

Thus $Nondom(B,f)$ includes all vectors being better than or incomparable with vectors from B. with respect to f.

3 A Dual Problem

With respect to the MOP (1) we can define a dual problem by 'inverting' the objective functions (replace *min* by *max*) and by considering only those alternatives which are infeasible for the original problem and in its dominance cone. The formal definition is as follows:

$$\text{"}max\text{"} f(y), \text{s.t.} y \in Dom(A,f). \tag{7}$$

Note that for any $y \in Dom(A,f)$ there exists $x \in A$ with $f(y) \leq f(x)$. Furthermore, we have that for any $x \in A$ there does not exist $y \in Dom(A,f)$ with $f(x) \leq f(y)$. Note that the efficient set for the dual problem can be defined analogously to (4). We the find the following result:

Duality Theorem:

Let x be an efficient solution of (1). Then x is also a dominating solution of (7). Let y be an efficient solution of (7). Then y is also a dominating solution of (1).

The proof is straightforward and omitted here for brevity. Note that according to the definition of the dual problem, there is no duality gap as often observed for dual pairs of MOPs defined according to different duality concepts (see [1]). For further and more general results on duality in MOP, see [11].

Usually, in Evolutionary Algorithms (EAs) we are dealing with populations of "entities" which correspond to feasible solutions. Let us assume for simplicity that feasible parent and offspring solutions are given as follows:

$$M^t = \{a_1^t, ..., a_\mu^t\} \subseteq A,$$

$$N^t = \{a_1^t, ..., a_\lambda^t\} \subseteq A,$$

M^t is the parent population in generation t which is assumed to consist of μ entities. N^t is the offspring population in generation t which is assumed to consist of λ entities.

During the run of a multiobjective optimization algorithm, $Dom(A, f)$ is, of course, not known. We only have an approximation of the efficient set based on a current population. $Dom(M^t, f)$ usually is too small to enclose the interesting part of the alternative space close to the efficient set of the original problem. In particular, areas of solutions being incomparable with all solutions in M^t would be missing. Therefore, we include these solutions and assume $y \in Nondom(M^t, f)$ for a dual optimization problem considered in generation t of the evolutionary algorithm.

4 The Algorithm PDMOEA

Our new algorithms adheres to the usual general framework of EAs which is similar to the following pseudo code. Note that the sequence of the steps 3–8 (also denoted as genetic operators), in particular mutation and recombination, may be different in other variants of evolutionary algorithms. Furthermore, there is no clear distinction between offspring and the parents of the next generation in some variants of evolutionary algorithms (such as standard genetic algorithms).

(a) Initialize starting population M^0.
(b) Initialize control parameters; t:=0.
(c) Copy and mutate N^t from M^t.
(d) Recombine N^t.
(e) Evaluate fitness of N^t and M^t.
(f) Select M^{t+1} from $N^t \cup M^t$.
(g) If stopping criterion fulfilled then stop.
(h) $t := t + 1$; goto 3.

Thus, an EA basically consists of a generational loop producing offspring solutions from parent solutions using some variation principles and selecting new parent solutions according to their fitness.

A more recent idea in dealing with infeasible solutions (which allow to calculate the objective function) is just to keep some of them in the population (or within a separate population of infeasible solutions) for further processing. Of particular interest are those infeasible solutions which dominate the Pareto front built by the feasible solution set from the current population or which explore new regions being incomparable (with respect to the Pareto relation) to existing solutions.

The advantages are quite clear:

- No waste of time for repair or recalculation of solutions
- Pareto set may be approximated from "both sides"
- Possibly faster/better approximation

Figure 1 shows (for the case of a biobjective minimization problem) an example of how a recombination between a feasible and an infeasible solution leads closer to the true Pareto set than a recombination between two feasible solutions. Note that, there is, however, no necessity that new solutions located 'between' two other solutions also have intermediate objective values. In Fig. 2, it is illustrated how feasible and infeasible solutions may approach the true Pareto front from two sides. It becomes evident that the probability of accepting new offspring solutions is much higher when those infeasible solutions are allowed which dominate existing ones (with respect to the dual problem) or which are incomparable to those already being accepted.

In [7] we have analyzed theoretical properties such as convergence for a multiobjective evolutionary algorithm based on a feasible and an infeasible population.

However, the best general advice with respect to infeasibility might be, that one should avoid it by using an appropriate encoding. Occasionally, more intelligent data structures may avoid the infeasibility of solutions at all.

In the following, we sketch a new multiobjective evolutionary algorithm suitable for approximating the efficient set of an MOP (1) or, respectively, (6). The novel

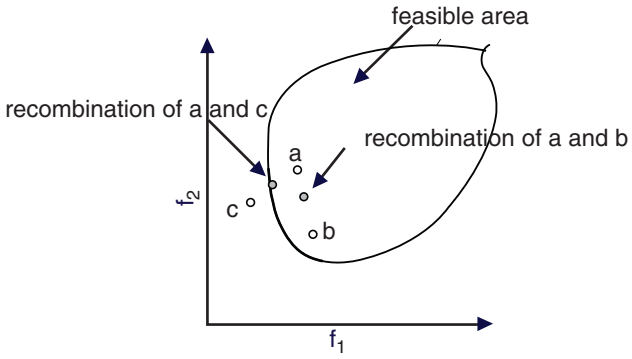


Fig. 1 Recombination between feasible and infeasible solutions

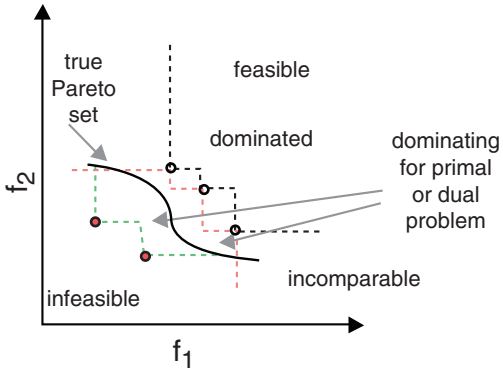


Fig. 2 Approximation of the Pareto front from two sides

concept of the primal–dual MOEA (PDMOEA) is to solve both problems, the original one and a dual problem, simultaneously. Therefore, we distinguish the parent and the offspring solution sets into feasible and nonfeasible solutions.

In the discussion of MOEAs, the specification of the operations for mutation and recombination are of smaller interest than the selection. This is because the multiobjective nature of the objective function only affects the fitness evaluation of a solution, and thus the selection step. Therefore, the discussion of other genetic operators and steps of the PDMOEA is omitted here for brevity. More details on the design of these operators can be found in [6].

With respect to the selection operator we favor an elitist selection which conserves earlier solutions as long as they remain efficient. In particular, such a selection allows for an analysis of theoretical properties of MOEAs such as convergence (see [4]). Since a population with a constant size fixed in advance is usually not capable of approaching the whole efficient set $E(A, f)$ with given exactness ε, we use an MOEA based on a variable population size. In [6] we have analyzed the convergence of such an approach being controlled by an 'approximation measure' ε, and which utilizes an increasable population.

For the PDMOEA we choose a similar concept allowing an increase of the population by new alternatives in order to support approximation. This is especially important since we assume that generating and evaluating new solutions might be computationally expensive (see [8] for an example) such that even infeasible alternatives might be worth keeping for approximating the efficient set from the 'infeasible side'.

Based on these considerations, we propose the following selection rules:

(a) Keep feasible parents which remain efficient.
(b) Add feasible offspring which dominates previously efficient parents.
(c) Add feasible efficient offspring which has a minimum distance of ε to all already selected parent and offspring alternatives.
(d) Keep parents which remain feasible and efficient for the dual problem.

(e) Add offspring being feasible for the dual problem which dominates previously efficient parents of the dual problem.
(f) Add dual efficient offspring which has a minimum distance of ε to all already selected dual feasible parent and offspring alternatives.

5 Some Computational Results: Tunneling the Infeasible Areas

For some computational tests we compare three variants of MOEAs using different strategies for coping with infeasibility. The first variant simply uses recalculation of a solution if the variation operators violates one or several constraints. The second variant uses a penalty approach. Each violation of a constraint leads to a correction of all objective function values which is accomplished by a penalty function. Basically, the penalty function is the amount of constraint violation multiplied by a factor. The third variant of handling infeasibility is the PDMOEA described above.

For computational experiments we have considered a multiobjective optimization problem described in [5], p. 356:

$$(P4) \qquad min f(x), f : R^2 \to R^2 \text{ with } f_i : x \mapsto x_i \text{ for } i \in \{1,2\}$$

and

$$x \in X = \{x \in R^2 : x_1 \geq 0, x_2 \geq 0, x_2 - 5 + 0.5 x_1 \, sin(4 x_1) \geq 0\}.$$

The problem (P4) is characterized by 7 curve segments of efficient solutions which are separated by infeasible areas. In such a case, it very much depends on the strength of the variation parameters (mutation rates, in particular) whether all these parts of the efficient set can be reached by the MOEA with a limited amount of time.

To make the results better comparable, each variant is allowed to use the same amount of time (1 second) on the same computer. Due to a different amount of calculations per generation and different resulting population sizes, the number performed generations varies: in the first case the MOEA runs for 2413 generations, in the second one for 2007 generations, and in the third one for 1189 generations. As can be seen, the PDMOEAs requires more computation time per generation. Nevertheless its results within the half no. of generations are significantly better than those of the other two approaches. The results of the three computer experiments (final feasible population in the objective space) are shown in Figs. 3–5. In these figures, also the curve resulting from the third restriction and defining parts of the Pareto front is shown.

For comparing the results, we do not present typical measures such as the coverage since these do not reflect the obvious preferability of some solution sets (see [13]). In fact, the variants of MOEAs show similar coverage results despite significantly different visualizations.

The final population of MOEA variant 1 only covers the first two upper left parts of the efficient set (see Fig. 3). Variant 2 which allows for a temporary infeasibility

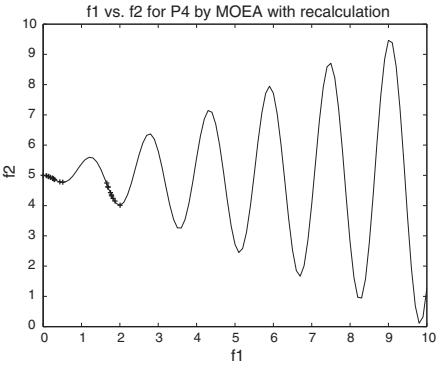


Fig. 3 Results for solving P4 using recalculations of infeasible solutions

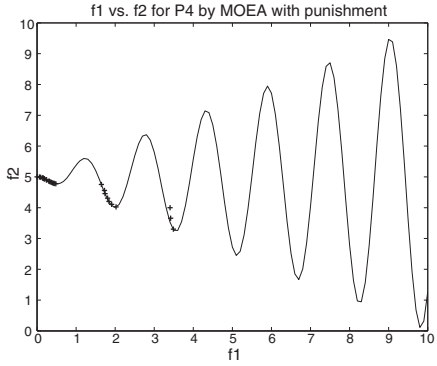


Fig. 4 Results for solving P4 using a punishment of infeasible solutions

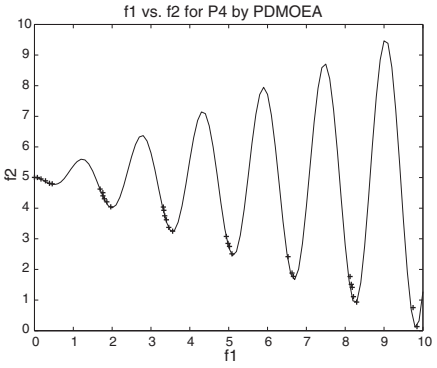


Fig. 5 Results for solving P4 using the new PDMOEA

due to the punishment concept reaches also part 3 of the efficient set (Fig. 4) while only variant 3 (the PDMOEA) reaches all of the disconnected efficient areas (Fig. 5).

This is due to some tunneling effect: Infeasible solutions are accepted for the population in those parts of the objective space which connect the separated parts of the efficient set. Figure 6 shows the final population resulting from the PDMOEA which includes solutions "walking" through the infeasible areas which separate the

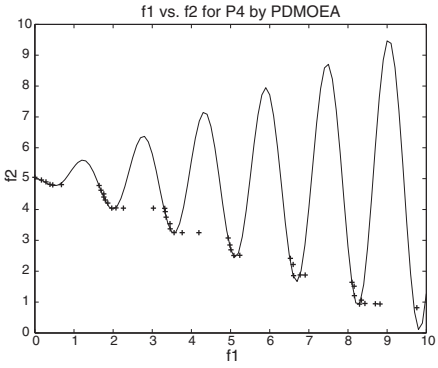


Fig. 6 PDMOEA solutions including infeasible ones

Pareto front. Note that the shown feasible solutions are not identical to those shown in Fig. 5 since the data is taken from a different stochastic repetition of the experiment. From ten stochastic repetitions of the experiment, the PDMOEA reached all of the disconnected efficient areas in nine cases.

6 Conclusions

The novel selection rules guarantee that we do not loose any solutions which are 'good' with respect to the original or a dual problem. An alternative in the parent population is replaced by an offspring alternative only if it is not any longer more efficient (for the primal or the dual problem). New efficient solutions for the primal or dual problem are added, if they are sufficiently distinct from already selected ones.

In this paper we presented first results of the novel way of treating infeasibility in comparison with two other approaches. Significant advantages were visible when the efficient set is not connected and variation operators have difficulties in bridging these gaps. The new approach allows in such cases for a tunneling of the infeasible domains separating the Pareto front.

Theoretical properties of the new algorithm have already been analyzed in a separate paper [7]. It will, however, be subject of a future study to compare the new algorithm with existing approaches using typical test problems.

References

1. Bigi G, Pappalardo M (2001) What is the duality gap in vector optimization? Technical Report TR-01-15, Università di Pisa, Italy
2. Deb K, Pratap A, Meyarivan T (2000) Constrained test problems for multi-objective evolutionary optimization. KanGAL Report No. 200002
3. Gal T (1986) On efficient sets in vector maximum problems - A brief survey. Eur J Oper Res 24:253–264

4. Hanne T (1999) On the convergence of multiobjective evolutionary algorithms. Eur J Oper Res 117(3):553–564
5. Hanne T (2000) Global multiobjectove optimization using evolutionary algorithms. J Heurist 6:347–360
6. Hanne T (2007) A multiobjective evolutionary algorithm for approximating the efficient set. Eur J Oper Res 176(3):1723–1734
7. Hanne T. (2007) A primal-dual multiobjective evolutionary algorithm for approximating the efficient set. Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, September 25–28, 2007
8. Hanne T, Nickel S (2005) A multi-objective evolutionary algorithm for scheduling and inspection planning in software development projects. Eur J Oper Res 167:663–678
9. Ishibuchi H, Kaige S, Narukawa K (2005) Comparison between Lamarckian and Baldwinian repair on multiobjective 0/1 knapsack problems. In: Lecture Notes in Computer Science 3410: Evolutionary Multicriterion Optimization - EMO 2005. Springer, Berlin, pp 370–385
10. Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs, 3rd rev. and extended edn. Springer, Berlin
11. Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic, Orlando
12. Schwefel H-P (1981) Numerical optimization of computer models. Wiley, Chichester
13. Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and applications. TIK-Schriftenreihe Nr. 30, ETH Zurich

The Effect of Initial Population Sampling on the Convergence of Multi-Objective Genetic Algorithms

Silvia Poles, Yan Fu, and Enrico Rigoni

Abstract This paper aims to demonstrate that the initial population plays an important role in the convergence of genetic algorithms independently from the algorithm and the problem. Using a well-distributed sampling increases the robustness and avoids premature convergence. The observation is proved using MOGA-II and NSGA-II with different sampling methods. This result is particularly important whenever the optimization involves time-consuming functions.

Keywords: Convergence · Initial population · MOGA-II, Multi-objective genetic algorithms · NSGA-II

1 Introduction

Genetic algorithms (GAs) are based on an analogy of natural selection and reproduction. Once an initial population has been randomly generated, the genetic algorithm evolves by means of its operators. This paper aims to show that the initial population plays an important role in the convergence and robustness of a genetic algorithm. Using a well-distributed sampling instead of a random initial generation, the alleles present in the initial group are diverse. A larger number of alleles implies gaining more information on the problem and so increases the robustness and the convergence of a generic genetic algorithm. Affenzeller in [1] demonstrated that if the genetic information present in the population is not enough, a genetic algorithm can suffer from premature convergence and get stuck in local optimal solutions.

S. Poles (✉)
ESTECO – Area Science Park - Padriciano, 99 – 34012 Trieste, Italy
e-mail: silvia.poles@esteco.com

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

Here we will prove the same observation using two different multi-objective genetic algorithms, MOGA-II [2] and NSGA-II [3] and comparing different sampling methods.

Several papers [4, 5] show that the construction of a well suited initial population can speed up GAs on specific problems and methods. We will prove that a well-distributed population speeds up the convergence to the correct Pareto front independently from the GAs and the problem; this result is particularly important whenever the optimization task involves time-consuming functions.

2 Initial Population Sampling

The initial population of a genetic algorithm can be generated using several types of sampling methodology. In this paper several different methods are studied and compared: Random, Sobol, Cross-Validation (Kriging), Latin Hypercube (LHS), Median LHS (MLHS) and Hammersley sequence (HSS) [6].

Random generates a sequence of random numbers by spreading points uniformly in the design space. It is a very simple method based on the mathematical theory of random number generation.

Sobol [7] is a deterministic algorithm (even known as a quasi-random sequence): its aim is to obtain a uniform sampling of the design space. With this method the clustering effects of random sampling are reduced, the points are maximally avoiding each other, making Sobol a low-discrepancy method.

Cross Validation distributes the designs uniformly in the design space, on the basis of the Kriging [8] algorithm used for response surfaces. In fact this method estimates the error of the model and then chooses a good new set of points in order to make the response surface more reliable.

Latin hypercube sampling (LHS) [9] is one form of stratified sampling that can reduce the variance in the Monte Carlo estimate of the integrand. The range of each input is divided into non-overlapping intervals of equal probability. One value from each interval is selected randomly with respect to the probability density in the interval. Then random pairing based on a pseudo-random number generator for all input variables are employed to formulate the final samples.

Median Latin hypercube sampling (MLHS) also divided each input variable range into non-overlapping intervals of equal probability, but only the median point of each interval is selected.

Hammersley sequence sampling technique [10] is another quasi-random number technique. It uses the Hammersley points to uniformly sample a $(k-1)$-dimensional hypercube, and the results revealed that the Hammersley points provide the optimal location for the sample points so as to obtain better uniformity in the $(k-1)$-dimension.

3 Test Cases

3.1 ZDT Problems

Algorithms are tested on three different two-objective optimizations problems, taken from the well-known ZDT functions [11]: ZDT1, ZDT2, ZDT3. These are unconstrained problems with two objectives to be minimized. The original problems have 30 variables ($n = 30$), but they are perfectly scalable by virtue of their formulation:

$$\begin{aligned}
\text{ZDT1}\quad & \min f_1(\mathbf{x}) = x_1 \\
& \min f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{x_1/g(\mathbf{x})}\right] \\
\text{ZDT2}\quad & \min f_1(\mathbf{x}) = x_1 \\
& \min f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - (x_1/g(\mathbf{x}))^2\right] \\
\text{ZDT3}\quad & \min f_1(\mathbf{x}) = x_1 \\
& \min f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{x_1/g(\mathbf{x})} - \tfrac{x_1}{g(\mathbf{x})}\sin(10\pi x_1)\right] \\
& \text{where} \quad x_i \in [0,1]\,, \quad i = 1,n \\
& \text{and} \quad g(\mathbf{x}) = 1 + 9\left(\textstyle\sum_{i=2}^{n} x_i\right)/(n-1)
\end{aligned}$$

The three problems are arranged in order of increasing difficulty (see Fig. 1): ZDT1 has a convex Pareto front, ZDT2 has a non-convex Pareto front, and ZDT3 has a disconnected Pareto front: it consists of five non-contiguous convex parts.

3.2 Summary Attainment Surfaces

In order to assess the performance of a stochastic multi-objective optimizer several runs (using different random seeds) are required. The outcome of each single run,

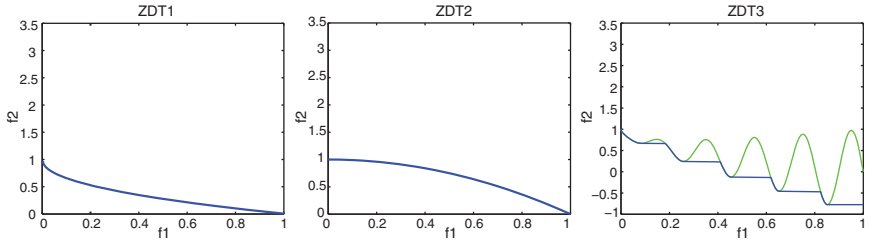


Fig. 1 Pareto fronts of the ZDT1, ZDT2, and ZDT3 problems

i.e. a Pareto set approximation, can be visualized by means of an attainment surface [12], defined as the family of tightest goals attained by the relevant approximation set. Summary attainment surfaces [13] are a suitable tool for studying the average behavior over many runs: in particular the median attainment surface is defined as the region that has been attained by the 50% of the runs.

3.3 Performance Metric

Several quality indicators or metrics have been proposed for performance assessment of multi-objective evolutionary algorithms (for example see [14]): the hypervolume indicator or $\mathscr{S}$ metric given in [15] is defined as the size of the region dominated by the relevant Pareto set approximation. The computation of this metric requires a suitable reference point (the worst point).

In this work the hypervolume metric is computed directly for the median attainment surface, and not for the Pareto set approximations of each single run. Furthermore, since the exact Pareto front is known for each ZDT problem, a normalized hypervolume metric can be introduced: it is the ratio between the area dominated by the relevant median attainment surface and the area dominated by the true Pareto front. So its value is always included in the interval $[0,1]$, and a value close to 1 means a good result. The reference point chosen for all the three problems is $(1.0,3.5)$: it corresponds to the upper-right corner of all the subsequent median attainment surfaces plots.

3.4 Preliminary Studies

A set of preliminary studies were conducted comparing the performance of MOGA-II and NSGA-II on some numerical two-objective minimization problems, starting from different initial populations. The numerical benchmarks used are the three ZDT functions with the problems dimension fixed to 30.

In order to make a fair comparison between the different initial population samplings with 100 points, we run the test functions again without changing the parameters for the GAs. So, apart from the test function, each optimization run has been re-executed 5 times using the following parameters for MOGA-II: 10 generations, directional cross-over probability 65%, classical cross-over probability 20%, selection probability 5%, mutation 10%, elitism enabled. The following parameters has been used for NSGA-II: 10 generations, crossover probability 90%, mutation probability for real-coded vectors 100%, distribution index for real-coded crossover 20, distribution index for real-coded mutation 20.

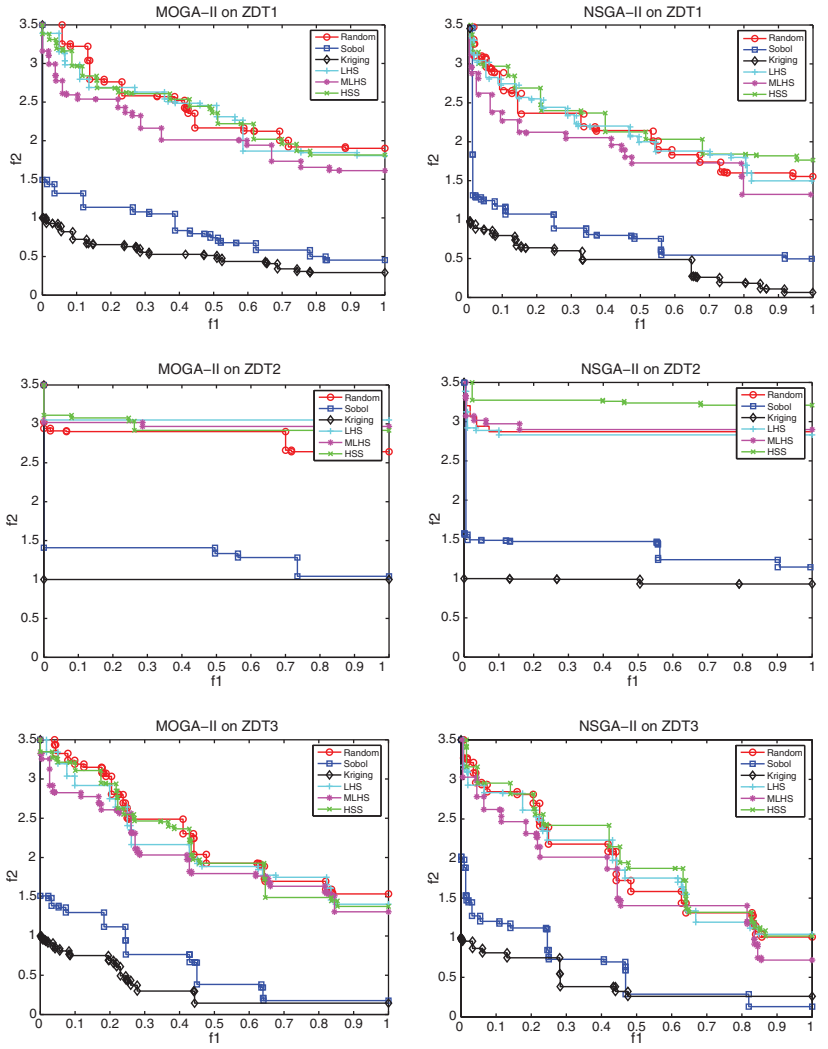


Fig. 2 Median attainment surfaces on ZDT functions; outcomes starting from *Kriging* and *Sobol* sequences always outperform other results regardless of the GA used

Even if the effect of initial population can strongly depend on mutation, this preliminary study was designed to investigate on the effect of the initial sampling on the convergence of GAs, so the mutation is kept fixed. Figure 2 is a median attainment surfaces plot summarizing some results. These results try out that a well-distributed initial population really speeds up the convergence to the correct Pareto front; this statement is true for both genetic algorithms. After ten generations, Sobol

and Kriging sequences are closer than a Random sequence to the Pareto front. Results of Fig. 2 has been obtained with an initial population of 100 points in order to respect the setting of [3].

These initial results confirm the intuition that a larger number of alleles in the initial population increases the robustness and the convergence of GAs. Anyhow, the impact of the sample size should be carefully considered; when the ratio of the population size and number of input variables is small, then uniformity and independence of the samples become more important.

3.5 Advanced Studies

In the advanced studied we sought to maximize the validity of our preliminary analysis by extending the tests. In particular, we reformulated the original problem by scaling the dimension from 2 up to 30. This approach allows to quantify the effect of dimension on the convergence rate. Furthermore we increased the number of generations, in order to allow a broader evolution and to study the effect of this factor. Since the preliminary study showed no qualitative difference between the two different genetic algorithms, here we limited the study to MOGA-II.

The experiment setup was the following: 3 test problems (ZDT1, ZDT2, and ZDT3), nine dimensions explored (2, 3, 4, 5, 10, 15, 20, 25, 30), three different degrees of evolution (10, 20, and 30 generations). With 20 and 30 generation we performed only the experiments with five or more dimensions, since with less dimensions all the samplings produce results very close to the real Pareto front and indistinguishable from each other.

In Fig. 3 the median attainment surfaces for the ZDT1 problem and for ten generation are presented: the results confirm the conclusions of the preliminary study. Kriging sampling produces the best results, followed by Sobol. The more the dimensions, more difficult the problem, and more evident the difference between the different samplings. Due to space limitations only the plots of the median attainment surfaces for the ZDT1 problem and for ten generation are presented. The behavior for ZDT2 and ZDT3 problems is qualitatively the same. With 20 and 30 generations the results are again qualitatively the same, with the obvious difference that all the approximation sets are closer to the real Pareto front, because of the more advanced evolution degree.

All of the tables of normalized hypervolume metric values are presented, for all the experiments performed, in Tables 1, 2, and 3. The relevant plots of performance metric versus problem dimensions are also shown in Figs. 4, 5, and 6. Some numbers in Tables 1, 2, and 3 may appear to be similar, but there is no need of statements about the statistical significance because all the charts demonstrate clearly the better performance of Kriging and Sobol, especially with higher dimensions, when the problem is harder.

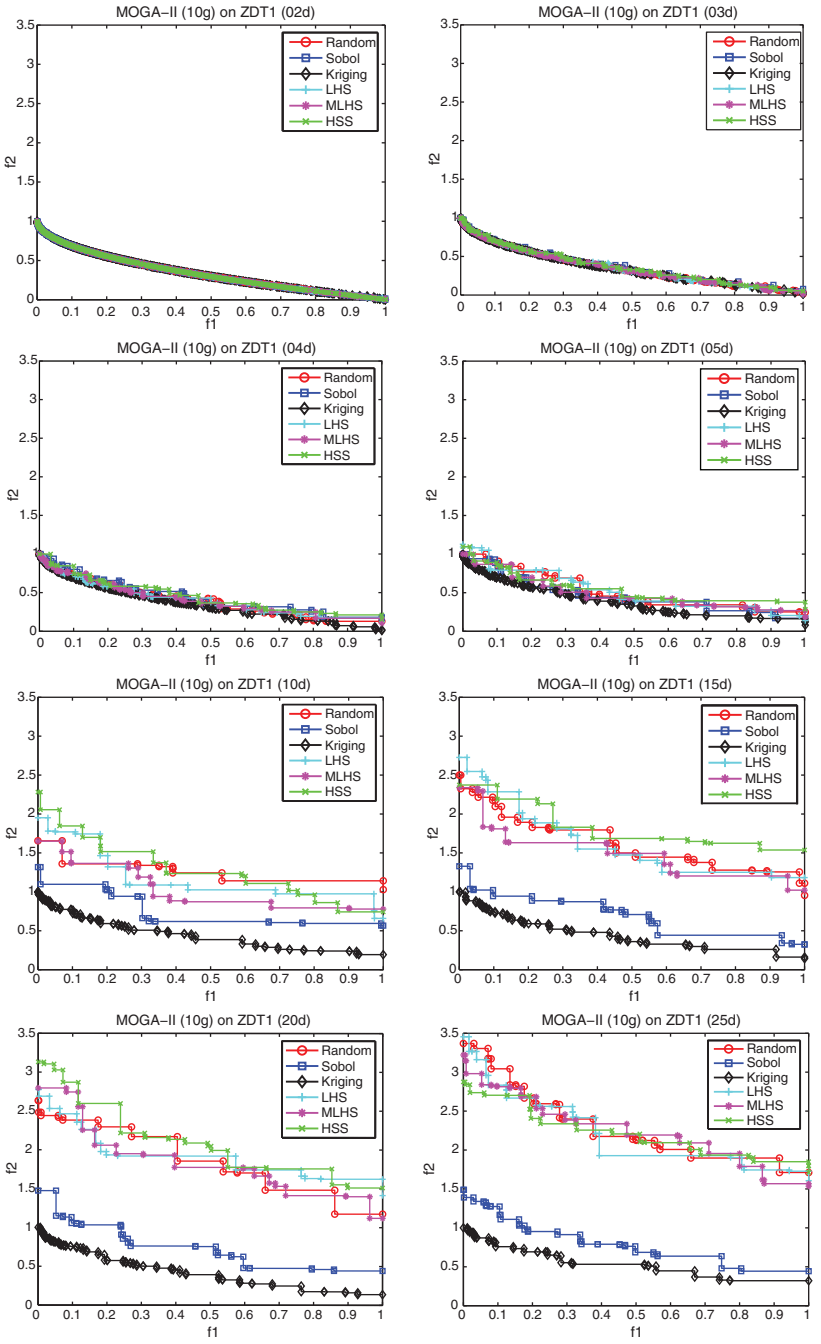


Fig. 3 Median attainment surfaces on ZDT1 function, varying the problem dimensions

Table 1 Normalized hypervolume metric tables for ZDT1 problem

ZDT1 - 10g									
d:	2	3	4	5	10	15	20	25	30
Random	0.9987	0.9944	0.9804	0.9437	0.7076	0.5940	0.5216	0.3862	0.3647
Sobol	0.9988	0.9886	0.9655	0.9605	0.8704	0.8870	0.8759	0.8618	0.8532
Kriging	0.9987	0.9950	0.9942	0.9846	0.9683	0.9673	0.9755	0.9399	0.9410
LHS	0.9985	0.9933	0.9772	0.9407	0.7341	0.5968	0.5017	0.4109	0.3647
MLHS	0.9987	0.9933	0.9762	0.9516	0.7808	0.6349	0.5261	0.3980	0.3608
HSS	0.9988	0.9886	0.9625	0.9342	0.7023	0.5288	0.4463	0.4074	0.3628

ZDT1 - 20g						
d:	5	10	15	20	25	30
Random	0.9964	0.9206	0.8601	0.7210	0.6629	0.6599
Sobol	0.9963	0.9748	0.9654	0.9548	0.9444	0.9231
Kriging	0.9985	0.9958	0.9957	0.9968	0.9914	0.9904
LHS	0.9952	0.9239	0.8047	0.7513	0.6252	0.5771
MLHS	0.9959	0.9524	0.8397	0.7546	0.6485	0.5830
HSS	0.9947	0.8989	0.8162	0.7257	0.7002	0.6205

ZDT1 - 30g						
d:	5	10	15	20	25	30
Random	0.9990	0.9790	0.9432	0.8770	0.8184	0.8021
Sobol	0.9986	0.9957	0.9871	0.9833	0.9768	0.9637
Kriging	0.9993	0.9987	0.9985	0.9987	0.9976	0.9975
LHS	0.9988	0.9752	0.9227	0.8903	0.7633	0.7202
MLHS	0.9990	0.9872	0.9421	0.8829	0.7964	0.7346
HSS	0.9987	0.9752	0.9154	0.8892	0.8360	0.7674

Table 2 Normalized hypervolume metric tables for ZDT2 problem

ZDT2 - 10g									
d:	2	3	4	5	10	15	20	25	30
Random	0.9981	0.9835	0.9664	0.9038	0.6671	0.5181	0.3620	0.2990	0.2383
Sobol	0.9978	0.9774	0.9326	0.9274	0.7575	0.8601	0.7590	0.7997	0.7820
Kriging	0.9970	0.9943	0.9911	0.9707	0.8855	0.8824	0.9059	0.9169	0.8824
LHS	0.9977	0.9876	0.9332	0.8661	0.7121	0.5980	0.3671	0.3113	0.2488
MLHS	0.9972	0.9930	0.9250	0.9035	0.6881	0.4353	0.3997	0.2613	0.1514
HSS	0.9978	0.9797	0.8928	0.9014	0.6707	0.4281	0.3766	0.1967	0.1903

ZDT2 - 20g						
d:	5	10	15	20	25	30
Random	0.9896	0.8900	0.7693	0.6104	0.5047	0.5062
Sobol	0.9898	0.8988	0.9379	0.8759	0.8931	0.8864
Kriging	0.9973	0.9880	0.9691	0.9685	0.9781	0.9295
LHS	0.9919	0.9069	0.8178	0.6755	0.6043	0.5110
MLHS	0.9847	0.8363	0.6957	0.6338	0.4830	0.4260
HSS	0.9782	0.8673	0.6950	0.6392	0.5479	0.4870

ZDT2 - 30g						
d:	5	10	15	20	25	30
Random	0.9979	0.9733	0.8763	0.7637	0.7178	0.6880
Sobol	0.9978	0.9699	0.9808	0.9339	0.9646	0.9309
Kriging	0.9989	0.9971	0.9969	0.9965	0.9954	0.9860
LHS	0.9987	0.9711	0.8768	0.7952	0.7327	0.6197
MLHS	0.9969	0.9447	0.8196	0.7782	0.6766	0.5702
HSS	0.9975	0.9479	0.8155	0.7631	0.6983	0.5872

Table 3 Normalized hypervolume metric tables for ZDT3 problem

ZDT3 - 10g									
d:	2	3	4	5	10	15	20	25	30
Random	0.9945	0.9786	0.9667	0.9210	0.7444	0.5771	0.5229	0.4093	0.3598
Sobol	0.9966	0.9843	0.9602	0.9046	0.8360	0.8636	0.8166	0.8237	0.8275
Kriging	0.9972	0.9843	0.9835	0.9702	0.9542	0.9261	0.9160	0.9004	0.9218
LHS	0.9967	0.9852	0.9457	0.9361	0.7333	0.5701	0.5013	0.4647	0.3924
MLHS	0.9963	0.9805	0.9493	0.9003	0.7175	0.5856	0.5724	0.4383	0.3638
HSS	0.9930	0.9814	0.9435	0.8826	0.6517	0.5028	0.4574	0.4038	0.3844

ZDT3 - 20g						
d:	5	10	15	20	25	30
Random	0.9942	0.9126	0.7936	0.7428	0.6164	0.5652
Sobol	0.9852	0.9557	0.9562	0.9193	0.9095	0.9058
Kriging	0.9942	0.9912	0.9769	0.9777	0.9727	0.9888
LHS	0.9913	0.9106	0.8181	0.6980	0.6415	0.6030
MLHS	0.9860	0.8946	0.8209	0.7574	0.6265	0.5342
HSS	0.9876	0.9209	0.7378	0.7238	0.6406	0.6239

ZDT3 - 30g						
d:	5	10	15	20	25	30
Random	0.9971	0.9776	0.9028	0.8560	0.7540	0.6925
Sobol	0.9911	0.9849	0.9825	0.9601	0.9511	0.9382
Kriging	0.9976	0.9946	0.9800	0.9915	0.9861	0.9943
LHS	0.9977	0.9714	0.9225	0.8201	0.7724	0.7680
MLHS	0.9950	0.9699	0.9101	0.8759	0.7584	0.6891
HSS	0.9960	0.9712	0.8765	0.8606	0.7783	0.7492

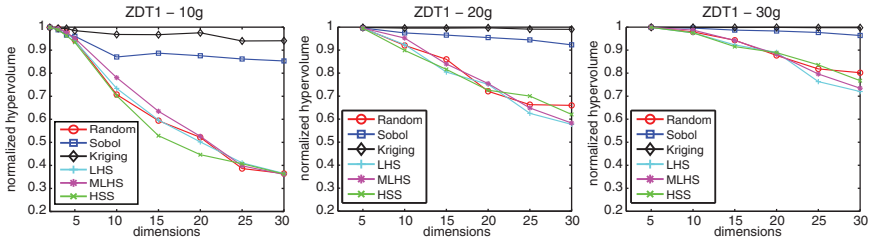


Fig. 4 ZDT1: normalized hypervolume metric versus problem dimension (for 10, 20, and 30 generations)

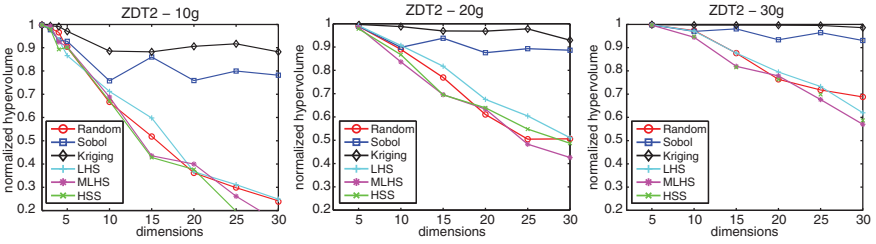


Fig. 5 ZDT2: normalized hypervolume metric versus problem dimension (for 10, 20, and 30 generations)

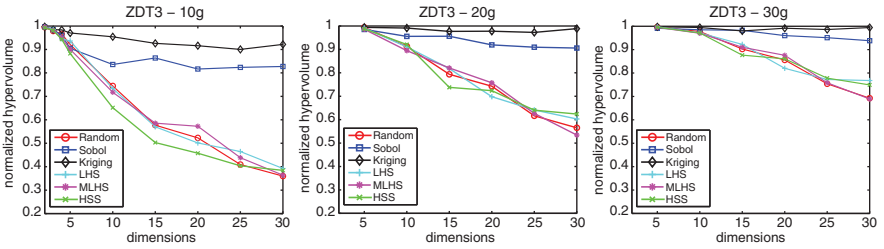


Fig. 6 ZDT3: normalized hypervolume metric versus problem dimension (for 10, 20, and 30 generations)

4 Results and Conclusion

The results of preliminary and advanced studies show that a well-distributed initial population really speeds up the convergence to the correct Pareto front: this statement is true for both genetic algorithms and all the three optimization problem being tested. The better performance of Kriging and Sobol sequences is more prominent in higher dimensions cases, when the problem is harder.

References

1. Affenzeller M, Wagner S (2004) The influence of population genetics for the redesign of genetic algorithm. J Syst Sci 30:41–49
2. Poles S (2003) MOGA-II: an improved multi-objective genetic algorithm. Technical report 2003-006, Esteco, Trieste
3. Deb K, Agrawal S, Pratab A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the Parallel Problem Solving from Nature VI Conference. Lecture Notes in Computer Science No. 1917, Paris, France, Springer, pp 849–858
4. Haubelt C, Gamenik J, Teich J (2005) Initial, population construction for convergence improvement of MOEAs. In: Evolutionary Multi-Criterion Optimization Proceedings of the Third International Conference, EMO 2005, Guanajuato, Mexico
5. Hill R, Hiremath C (2005) Improving genetic algorithm convergence using problem structure and domain knowledge in multidimensional knapsack problems. Int J Oper Res 1(1/2):145–159
6. Hammersley JM (1960) Monte Carlo methods for solving multivariable problems. Ann N Y Acad Sci 86:844–874
7. Sobol IM (1979) On the systematic search in a hypercube. SIAM J Numer Anal 16(5):790–793
8. Oliver MA, Webster R (1990) Kriging: a method of interpolation for geographical information systems. Int J Geogr Inform Syst 4(3):313–332
9. Iman RL, Shortencarier MJ (1984) A FORTRAN77 Program and User's Guide for Generation of Latin Hypercube and Random Samples for Use with Computer Models. NUREG/CR-3624, SAND83-2365, Sandia National Laboratories, California
10. Kalagnanam JR, Diwekar UM (1997) An Efficient Sampling Technique for Off-line Quality Control. Technometrics 39(3):308–319

11. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput J 8(2):173–195
12. Fonseca CM, Fleming PJ (1996) On the performance assessment and comparison of stochastic multiobjective optimizers. In: Voigt HM, Ebeling W, Rechenberg I, Schwefel HP (eds) Parallel problem solving from nature - PPSN IV , vol. 1141 of Lecture Notes in Computer Science. Springer, Berlin, pp 584–593
13. Knowles J (2005) A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications (ISDA V), Wroclaw, Poland
14. Fonseca CM, Knowles J, Thiele L, Zitzler E (2005) A tutorial on the performance assessment of stochastic multiobjective optimizers. Tutorial at EMO 2005
15. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms - A comparative study. In: Voigt HM, Ebeling W, Rechenberg I, Schwefel HP (eds) Parallel problem solving from nature - PPSN IV, vol. 1141 of Lecture Notes in Computer Science. Springer, Berlin, pp 292–301

Pattern Mining for Historical Data Analysis by Using MOEA

Hiroyuki Morita and Takanobu Nakahara

Abstract In data mining, graph mining is a promising new approach and some algorithms are proposed. However, their application is limited in the field of business. This is because of the wide diversity of business data. In this paper, we propose a method which extracts new valuable patterns by using graph mining approach and MOEA. In our method, historical purchasing data for each customer is transformed into tree structured data and gene is constructed from the structured data at first. Then the patterns are extracted by using existing MOEA from these genes. We apply our proposed method to a practical business data. From computational experiments, we show that our method has good performance and is able to extract valuable patterns from the view of business.

Keywords: Business data · Data mining · Graph mining · MOEA · Tree structured data

1 Introduction

Graph mining is a promising new approach in data mining research. It was developed and focused mainly on the sophisticated fields of organic chemistry and web analysis. Further, some efficient algorithms were proposed in the last decade [1, 5, 7, 10, 11]. The study of algorithms in this field is categorized into two groups– research to find effective subgraphs from a collection of general graphs and research to count frequent patterns in a tree graph. The algorithms in the former group are AGM [5], FSG [7], and gSpan [10], and the algorithms in the latter group are FREQT [1] and TreeMiner [11]. Their performance in the existing applications is efficient; however, their application in the field of business is limited. This is because of the wide diversity of business data. For example, the existing algorithms

H. Morita (✉)
Economics, Osaka Prefecture University, 1-1, Gakuen-cho, Sakai, Osaka, Japan
e-mail: morita@eco.osakafu-u.ac.jp

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

are effectively used to identify molecules in chemical compounds. In organic chemistry, the number of atoms represented by nodes in a graph is few; therefore, the size of the graph is restricted. However, business data is diverse and large. Thus, the direct application of the existing algorithms to this data is expensive in terms of the computational cost involved.

Data mining related problems include difficult combinatorial optimization problems; therefore, many evolutionary algorithms have been utilized since the 1990s. Cantu-Paz and Kamath [2] categorized these studies into four groups-feature extraction, feature selection, classification, and clustering. Several studies were surveyed for this categorization; however, most of them treated single objective optimization problems. Recently, some researchers have proposed multiobjective approaches for the problem presented in [3,6,8,9]. We believe that these are positive approaches because these data mining problems have various criteria to be fulfilled. In the business field, in particular, it is important to evaluate the data mining problems from multiple points of view. The previous studies have proposed MOEA approaches such as rule extraction and decision tree analysis using given explanatory variables for the existing mining problems. The studies show interesting results for each application as an extension of the existing studies.

In our study, we propose a method that extracts new patterns not by combining the given variables but by creating new ones from historical purchasing data by using a tree structured data and a multiobjective evolutionary algorithm (MOEA). Moreover, some wild cards are permitted in the patterns extracted. This enables the mining of more flexible patterns for business applications. In order to realize our idea, our algorithm comprises two steps. The first step is transforming the historical purchasing data* into tree structured data. We refer to the structured data as a "historical tree" (HT). The second step is finding effective patterns from the HT by using an MOEA. Our MOEA is based on an existing one [4]. Finally, a decision tree model is generated by using these patterns as new explanatory variables. Computational experiments indicate that our MOEA can find many promising patterns that will construct a decision tree model, which is better than the existing methods.

The remainder of this paper is organized as follows. Section 2 proposes a method to transform historical purchasing data into a tree structured data. Section 3 explains our MOEA. Section 4 reports the experimental results. Section 5 constructs the decision tree model using extracted patterns. Finally, Concluding remarks are discussed in Section 6.

2 Historical Purchasing Data and Historical Tree

Here, we use historical purchasing data of credit card purchases made by customers over a period of 2 years. Customer attributes such as gender, date of birth, and occupation, are provided. Although the payment amount and the means of payment are

* The data is obtained from the Data Analysis Competition 2005, which was sponsored by Joint Association Study Group of Management Science (JASMAC).

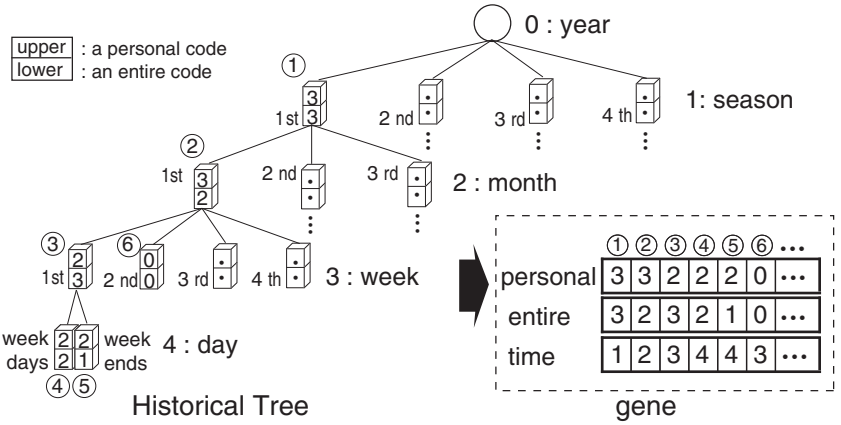


Fig. 1 Historical tree and gene

provided, the items purchased by the customers are unclear. In general, the customer attributes and the amount paid for some terms are used as explanatory variables to construct models. In our method, the historical purchasing data is transformed into a tree structured data. Further, patterns are extracted from the structured data and are used as additional explanatory variables to construct a decision model.

Figure 1 illustrates a historical tree (HT). The time levels for a HT vary from 1 year to 1 day. This is because we analyze data during 1 year, as mentioned later. Further, the time level of the top node, that is, the root node, is a year. However, there is only one node associated with this time level. Due to this the root node does not have a value. In the data, the daily purchase for majority of customers was zero; this is because credit cards are not commonly used on a daily basis in Japan. Therefore, we use two time intervals –weekdays and weekends– instead of the one day time interval in order to avoid redundancy in the mining patterns. According to the time levels, the amount paid by each customer is recorded. At each time level, two codes are mapped according to the amount paid-one is a personal code and the other is an entire code. The personal code represents the amount paid by each customer as compared to the distribution of the amount for each customer at each time level. Conversely, the entire code represents the amount paid by each customer as compared to the distribution of the amount of all the customers at each time level. Two types of codes are used because the distribution of purchasing is different for each customer. Evidently, the average purchases are greater among the rich customers as compared to the ordinary customers. Therefore, these two codes depict the changes in individual purchases and their impacts on the entire purchasing.

Each HT represents a purchasing pattern for a customer. Some partial graphs of HT are common for a set of customers; further, a section of the HT- is characteristic for a particular customer. Of these, a subgraph, which is common only for a specific set of customers, is useful for identifying the customers. In the next section, we extract such patterns by using the MOEA.

3 Pattern Mining for *HT* Using MOEA

In order to apply the MOEA to the HT, we initially transform it into a gene. The method employed is to trace each node from the root node by the order of a depth first search. Both the personal codes and the entire codes are transformed in the same manner. Further, for each code a time code is added to these two codes. The time code represents the position of the time level that is common to both codes; thus, season, month, week, and weekdays and weekends are denoted by codes 1, 2, 3, and 4, respectively. If this code is not included, it is difficult to decode an extracted pattern into a subgraph of the HT. As a whole, each HT is transformed into a triple gene with length 172.

Among the customers who use credit cards, the number of customers with revolving credits is less in Japan. Owing to this, Japanese credit corporations try to encourage customers in order to increase their profits in the future. Here, we define two sets of customers-*LR* and *L*. Although both make only lump-sum payments during the first year, in the second year, the mode of payment for *L* does not change, and *LR* adopts revolving credit. In other words, *LR* and *L* perform similarly in terms of the mode of payment during the first year; however, both exhibit a gap in the mode in the second year. The differences observed between the sets of customers in the data of the first year are interesting for business applications. Our objective is to find effective patterns that identify *LR* from these two groups.

Given a pattern $p \in P_l$ and where P_l denotes a set of patterns with length l , LR_p and L_p represent the sets of customers with p for each set *LR* and *L*, respectively. Then, each support, which is a ratio of the customers with p to all customers in *LR* and *L*, is defined as follows:

$$SUP(LR, p) = \frac{|\, LR_p \,|}{|\, LR \,|}, \tag{1}$$

$$SUP(L, p) = \frac{|\, L_p \,|}{|\, L \,|}, \tag{2}$$

where $|\cdot|$ denotes the number of the set. In order to extract effective patterns, one support value is should be larger than the other because the characteristic patterns are desirable in only one set. Therefore, we have to optimize the following two bi-objective problems:

$$(P1) \begin{cases} \text{maximize } SUP(LR, p) \\ \text{minimize } SUP(L, p), \end{cases} \tag{3}$$

$$(P2) \begin{cases} \text{maximize } SUP(L, p) \\ \text{minimize } SUP(LR, p). \end{cases} \tag{4}$$

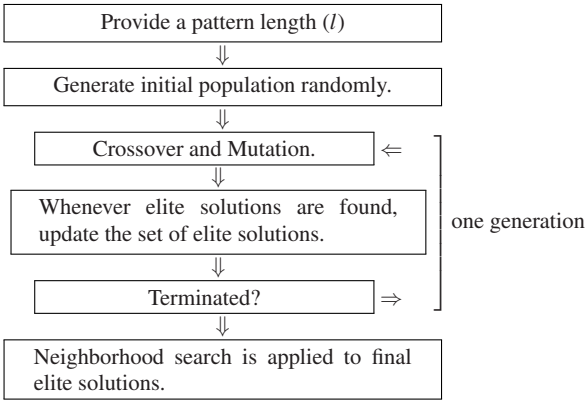


Fig. 2 Flowchart of our algorithm

Given a pattern with length l, two bi-objective pattern mining problems are solved by using MOEA. Using the basic ideas of [4], some sections are modified to adapt to the problem. Figure 2 illustrates the flowchart of our algorithm.

Firstly, pattern length l is provided. From our preliminary experiments, we observed that it is difficult to find effective patterns from the data when l is larger than 13. Therefore, we assigned values from 2 to 12 to l and found patterns for each l. After generating 200 initial solutions randomly, a crossover operation and a mutation operation are performed 150 times and 100 times, respectively, at every generation. When an initial solution is generated, the values of the personal code and the entire code are chosen from candidate values. With regard to the time code, the front position is chosen randomly and the following time codes are chosen from the permutation of the original time code. In general, the number of kinds of time code has relationship with the computational cost. Moreover, l has some influence on them too. From some preliminary experiments, we determined sufficiently large values for l.

In the crossover operation, we cannot apply the existing methods directly. In our MOEA, we split a triple gene into three single genes- the personal code, the entire code, and the time code. For each personal code and entire code, the existing crossover method (e.g., one point crossover method) is performed independently. For the time code, two types of genes from the parent are duplicated in order to maintain the feasibility of new solutions. Each code has two types of candidate single genes; thus, 2^3 new offsprings are generated by combining these genes, as shown in Fig. 3. During the crossovers and mutations, every elite solution is retained, and the solutions dominated by an elite solution are removed from the population. If no new elite solution is found for 100 consecutive generations, the evolutionary algorithm process is terminated. Finally, a neighborhood search is performed on all final elite solutions to improve the performance. In the search, neighborhood solutions are generated by swapping two any codes of the personal code and the entire code. Any new elite solution updates the set of elite solutions.

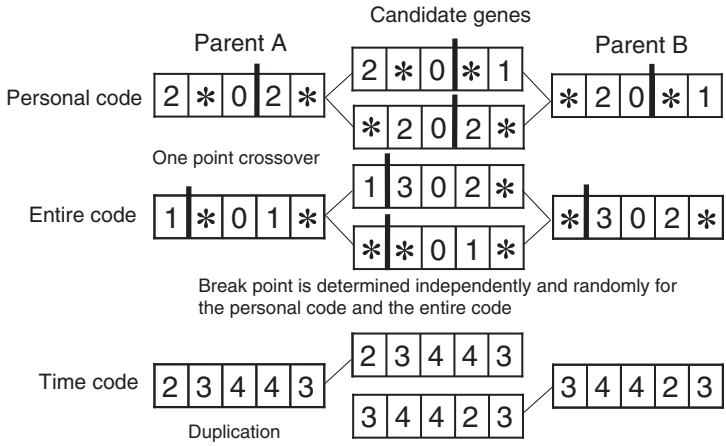


Fig. 3 Crossover method for our experiments

In the patterns, we permit a wild card that is equal to any of the codes- personal code or entire code. In business applications, it is not always necessary that all the codes of two patterns are identical. The partial difference between the codes is interpretable and useful; however, too many wild cards generate meaningless patterns. In our experiments, the number of wild cards used is restricted to l. This is half the number of the personal and entire codes.

4 Computational Experiments

The computational experiments are performed on a Pentium M 1.5 MHz with 512 MB RAM. The programs are coded in C language. Figure 4 illustrates an example of the final elite solutions ($l = 7$). The upper left shows the elite solutions for $P1$, and the lower right shows the elite solutions for $P2$.

In order to validate the performance of our MOEA, we compare our solutions with the pareto solutions. We can find all the pareto solutions upto $l = 4$ by enumeration. Figure 5 plots the pareto solutions and our approximate solutions. For $P1$ and $P2$, 49 and 70 pareto solutions exist, and the detection ratios of our approximations are 73.47 and 81.43%, respectively. Although some approximate solutions found are not exact, we observe that alternative solutions are found near them. On the whole, our solutions have a good approximation.

The average computational time for each l is shown in Fig. 6. In the experiments, data of 12,160 customers are used as input data. From the figure, the computational time greatly depends on l. Although a large computational cost is needed for $l = 11$, it is completely acceptable because input data is sufficiently large.

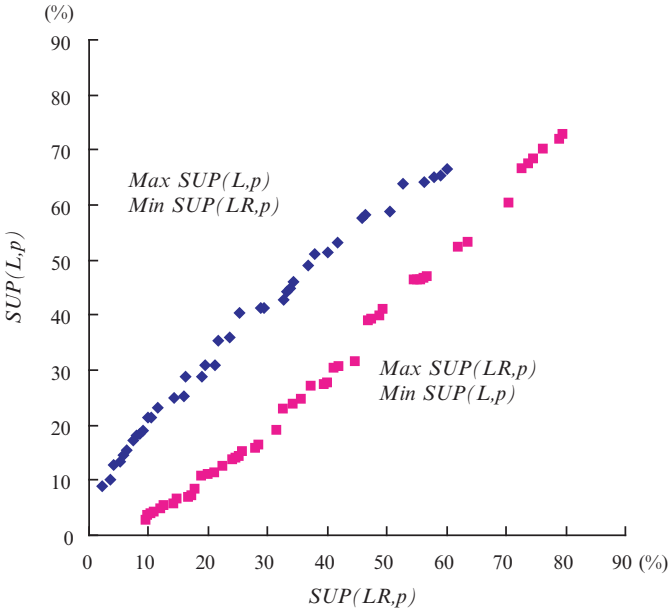


Fig. 4 Example of pareto solutions ($l = 7$)

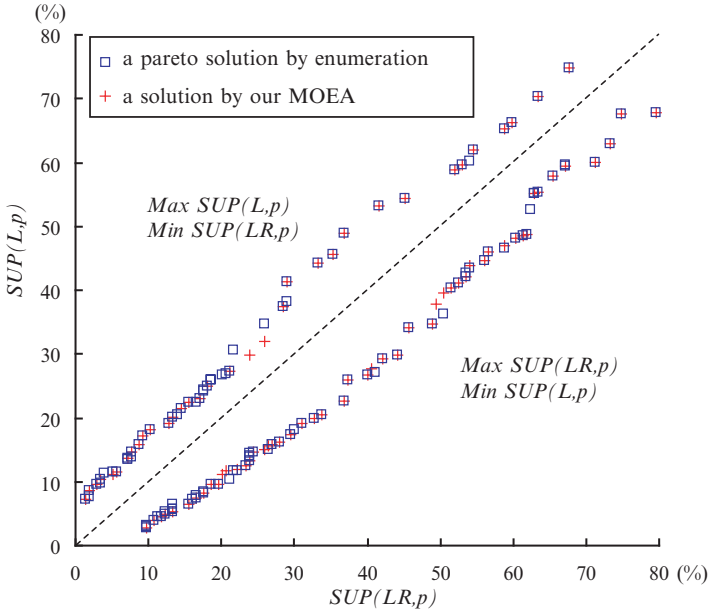


Fig. 5 Comparing our solutions with pareto solutions ($l = 4$)

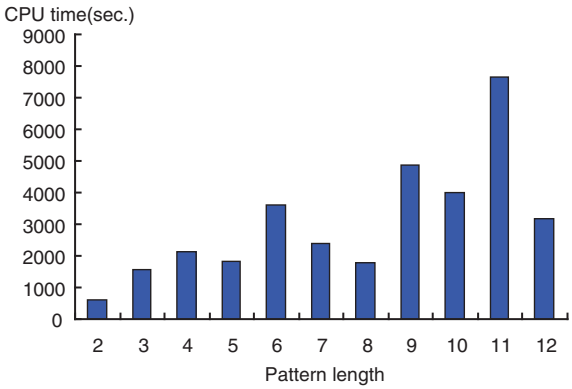


Fig. 6 Average CPU time to find approximate solutions

From the computational experiments, 492 promising patterns are observed to identify *LR* customers. By using these patterns, a decision tree is constructed in the next section.

5 Decision Tree Using the Patterns

Here, the identification of *LR* or *L* is an objective variable, and the patterns extracted and the given attributes of the customer are used as explanatory variables. Since $|LR|$ and $|L|$ are identical after sampling, a decision tree is constructed. Figure 7 shows this tree. In this figure, $\boxed{LR}$ and $\boxed{L}$ denote leaves that are identified as the expected group. A numerical value in parenthesis under a leaf denotes the accuracy of the discrimination. The entire accuracy of this model is 66% and it starts from 50%. Compared to the ordinary decision tree model that does not use patterns as explanatory variables, we can improve the accuracy of discrimination by about 10%.

In the figure, three patterns are utilized as strong explanatory variables. In the patterns, "$*$" denotes a wild card in the personal codes and the entire codes. Further, the characters in the time code denote each time level as shown in the figure.

Pattern 1 shows a characteristic pattern of a normal amount of purchasing during a month, and the purchasing that occurred during the first week of the month but not during the second week. The customers having this pattern are identified as *LR* customers. However, pattern 2 shows that normal amount of purchasing that exists is similar to pattern 1; however, there is no purchasing during the first week of the month and there is a normal amount of purchasing during last week of the previous month. The customers having this pattern are identified as *L* customers. Pattern 3 is similar to pattern 1; therefore, customers having that pattern are identified as *LR* customers. As a result, these patterns show that purchasing during the first week of a month introduces customers to *LR*, and purchasing during the last week of a month

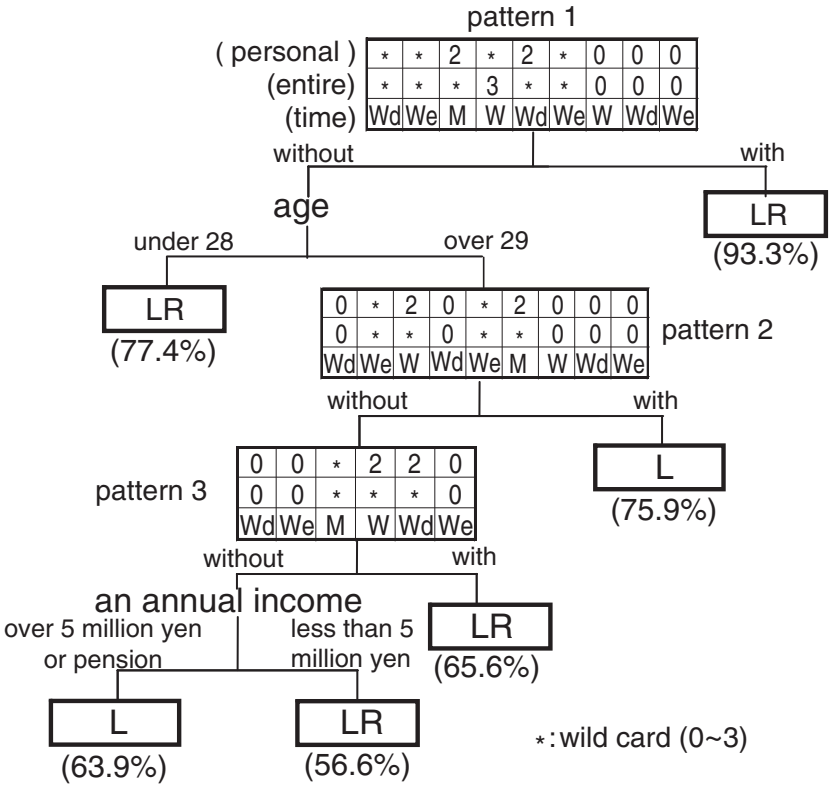


Fig. 7 Decision tree using patterns we found

introduces them to *L*. This is because of a gap between the account day and the payment day of credit cards. The account day is the last day of month and the date of payment is a day in the following month. Based on this, if the customers approve of pattern 1 or pattern 3, they can delay their payments. When such customers strongly desire to buy an item, they are expected to shift from being *L* customers to *LR* customers.

6 Concluding Remarks

In this paper, we propose a method to mine effective patterns in terms of multiobjective optimization from historical purchasing data by using MOEA. Considering the computational experiments on the historical purchasing data in practice, our method can find several promising patterns. Further, the accuracy of a decision tree model is improved by about 10% as compared to the ordinary method.

As future research, we could apply our method to several various historical purchasing data to obtain a more general method. With regard to MOEA, three extensions are promising. One of them is the extension of a triple gene, discussed in this paper, to more multiple genes. This extension implies that it is possible to mine patterns of a number of categories of a general POS data simultaneously. Another extension is the improvement of our MOEA to mine patterns without specifying the pattern length. The improvement of generating initial solutions from the elite solutions using the previous pattern length proceeds to the next implementation automatically. The last extension is the removal of numerous redundant patterns. Numerous redundant patterns are generated within a problem and among problems because several wild cards are permitted. Although these patterns are useful for finding new elite solutions, it is redundant to construct a decision tree model using them. After implementing MOEA, a filter method will be needed for all the patterns extracted in order to construct a decision tree model efficiently.

References

1. Asai T, Abe K, Kawasoe S, Arimura H, Sakamoto H, Arikawa S (2002) Efficient substructure discovery from large semi-structured data. In: Proceedings of SIIM on Data Mining (SDM2002), pp 158–174
2. Cantu-Paz E, Kamath C (2002) On the use of evolutionary algorithms in data mining. In: Abbass H, Sarker R, Newton C (eds) Data mining: a Heuristic approach. IDEA Group, Hershey, PA, pp 48–71
3. Fu Z, Golden B, Lele S, Raghavan S, Wasil E (2003) Genetically engineered decision trees: population diversity produces smarter trees. Oper Res 51(6):894–907
4. Gandibleux X, Morita H, Katoh N (2001) The supported solutions used as a genetic information in a population heuristic. 1st International Conference on Evolutionary Multi-Criterion Optimization (EMO2001), Lecture Notes in Computer Science, vol 1993. Springer, Berlin, pp 429–442
5. Inokuchi A, Washio T, Motoda H (2000) An apriori-based algorithm for mining frequent substructures from graph data. In: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, Lecture Notes in Computer Science, vol 1910. pp 13–23
6. Ishibuchi H, Yamamoto T (2004) Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy Sets Syst 141(1):59–88
7. Kuramochi M, Karypis G (2001) Frequent subgraph discovery. In: Proceedings of IEEE International Conference on Data Mining (ICDM2001), pp 313–320
8. Malard JM (2005) A role for pareto optimality in mining performance data. Comput Concurrency Practice Experience 17:1–21
9. Romero C, Ventura S, Bra PD (2004) Knowledge discovery with genetic programming for providing feedback to courseware authors. UMUAI 14:425–464
10. Yan X, Han J (2002) gSpan: Graph-based substructure pattern mining. In: Proceedings of IEEE International Conference on Data Mining (ICDM2002), pp 721–724
11. Zaki MJ (2002) Efficiently mining frequent trees in a forest. In: Proceedings of Special Interest Group on Knowledge Discovery and Data mining (SIGKDD2002), ACM, New York, pp 71–80

Multiple-Objective Genetic Algorithm Using the Multiple Criteria Decision Making Method TOPSIS

Máximo Méndez, Blas Galván, Daniel Salazar, and David Greiner

Abstract The so called second generation of Multi-Objective Evolutionary Algorithms (MOEAs) like NSGA-II, are highly efficient and obtain Pareto optimal fronts characterized mainly by a wider spread and visually distributed fronts. The subjacent idea is to provide the decision-makers (DM) with the most representative set of alternatives in terms of objective values, reserving the articulation of preferences to an a posteriori stage. Nevertheless, in many real discrete problems the number of solutions that belong the Pareto front is unknown and if the specified size of the non-dominated population in the MOEA is less than the number of solutions of the problem, the found front will be incomplete for a posteriori Making Decision. A possible strategy to overcome this difficulty is to promote those solutions placed in the region of interest while neglecting the others during the search, according to some DM's preferences. We propose TOPSISGA, that merges the second generation of MOEAs (we use NSGA-II) with the well known multiple criteria decision making technique TOPSIS whose main principle is to identify as preferred solutions those ones with the shortest distance to the positive ideal solution and the longest distance from the negative ideal solution. The method induces an ordered list of alternatives in accordance to the DM's preferences based on Similarity to the ideal point.

Keywords: Multi-objective evolutionary algorithm · 0–1 Multi-objective knapsack problem (0–1MOKP) · Multiple criteria decision making · Preferences · Safety systems design optimisation · TOPSIS

M. Méndez (✉)
Computer Science Department, University of Las Palmas de Gran Canaria Edif. de Informática y Matemáticas. Campus de Tafira, CP 35017 Las Palmas, Spain
e-mail: mmendez@dis.ulpgc.es

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

1 Introduction

Many well known and extendedly used Multi-Objective Evolutionary Algorithms (MOEAs) like NSGA-II [2] pursue to reach the efficient frontier and to sample it by a wide and even distributed set of non-dominated solutions. Subsequently, the decision maker (DM) chooses one solution in accordance with his/her preferences. Nevertheless, this approach does not always turn out the most appropriate. For instance, in many real discrete problems, if the efficient set is numerous and the size of the non-dominated population is limited, the MOEA cannot contain the whole set of solutions, compelling the DM to lose potentially attractive alternatives. In order to solve the abovementioned disadvantage, a possible strategy is to concentrate the search in a smaller set of Pareto optimal solutions, according to some DM's preferences. The incorporation of preferences into a MOEA is not new [1,3]. Nonetheless, to the best of our knowledge there is no previous attempt at incorporating TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) method [6] into a MOEA. In that sense we propose TOPSISGA, combining the concept of TOPSIS (minimizing the distance to the ideal solution while maximizing the distance to the negative solution) with MOEA methodologies (we use NSGA-II). With TOPSIS, a DM needs input his/her preferences or weights that are used in the proposed method for guiding the search towards the region of interest. The method induces an ordering of the solutions based on Similarity to the ideal point.

2 TOPSIS Method

The TOPSIS method was developed by Hwang and Yoon [6] for solving MCDM problems with a finite number of solutions. The TOPSIS method establishes that the chosen solution should have the shortest distance to the positive ideal solution (I^+) and the longest distance from the negative ideal solution (I^-), where the distances are calculated with a particular value of p ($1 \leq p \leq \infty$) of the Minkowski's metrics $L_p = \left\{ \sum_{i=1}^{k} w_i^p \, |f_i(\overrightarrow{x}) - f_i^*|^p \right\}^p$. Here, f_i^* $(i \in \{1,2,...,n\})$ is a vector whose coordinates corresponds to the coordinates of a reference point. With the TOPSIS method, that point is I^+ or I^-. The TOPSIS concept is rational and comprehensible. Since the Minkowski's metrics are weighted distances, the order strongly depends on the weights the DM assigns to each objective according to their preferences. The TOPSIS procedure consists of:

Step 1. Obtain a decision matrix, where a set of alternatives (solutions) $A = (a_j,$ $j = 1,2,\ldots,k)$ is compared with respect to a set of criterion functions (objective functions) $C = (c_i,\ i = 1,2,\ldots,n)$, an element x_{ij} of the matrix, is a value indicating the performance rating of jth alternative with regard to the criterion c_i.

Step 2. Calculate the normalized decision matrix according to:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{k} x_{ij}^2}}.$$

Step 3. Calculate the weighted normalized values as:

$$v_{ij} = w_i r_{ij}$$

w_i is the weight of the ith criterion set by the DM and $\sum_{i=1}^{n} w_i = 1$.

Step 4. Determine the positive ideal solution I^+ and the negative ideal solution I^- as:

$I^+ = (\max_j v_{1j}, \max_j v_{2j}, ..., \max_j v_{nj}) = (v_1^+, v_2^+, ..., v_n^+)$, see Fig. 1(a)

$I^- = (\min_j v_{1j}, \min_j v_{2j}, ..., \min_j v_{nj}) = (v_1^-, v_2^-, ..., v_n^-)$, see Fig. 1(a).

Step 5. Calculate the Euclidean distances for each alternative from the positive ideal solution as:

$$d_j^+ = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^+)^2} \qquad \text{j=1,2,...k.}$$

Similarly, the Euclidean distances from the negative ideal solution is given as:

$$d_j^- = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^-)^2} \qquad \text{j=1,2,...k.}$$

Step 6. Calculate the relative closeness to the positive ideal solution (rating of Similarity to the ideal positive) as:

$$D_j^+ = \frac{d_j^-}{d_j^+ + d_j^-} \qquad D_j^+ \text{ value [0,1].}$$

Step 7. Sort the solutions in terms of similarity. The final (increasingly labelled) order is obtained sorting the set of alternatives decreasingly in terms of D_j^+, i.e. from the most similarity to the less. Figure 1(b) shows the basic principle, a_j is closer to the positive ideal and farther from the negative ideal than a_z because $d_j^+ < d_z^+$ and $d_j^- > d_z^-$; $D_j^+ > D_z^+$ and the alternative a_j is better than a_z.

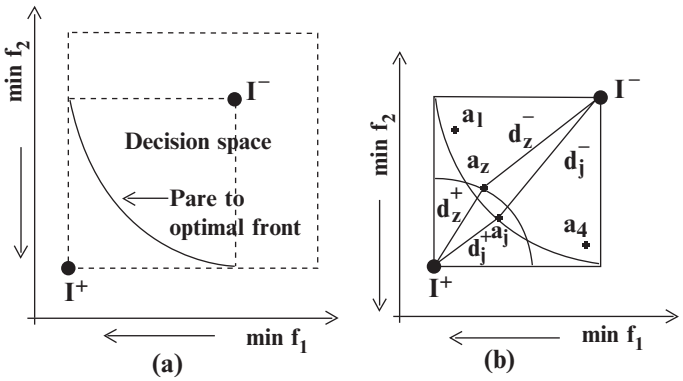


Fig. 1 (**a**) Positive and negative solutions, (**b**) TOPSIS distances

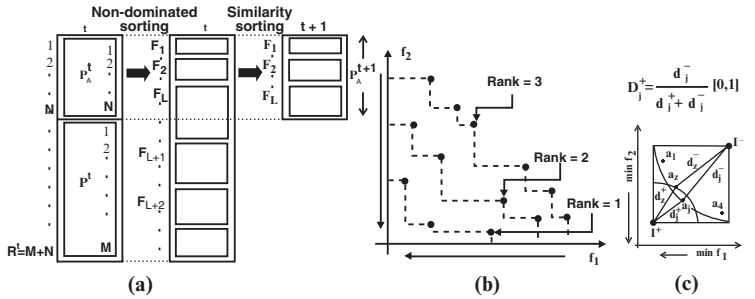


Fig. 2 Rank (**a**) (**b**) and similarity (**c**) concepts used by TOPSISGA

3 Proposed TOPSISGA Method

The present approach has a similar structure to other MOEAs, and it introduces two modifications to the original formulation of NSGA-II [2]: First the size of population and the archive of non-dominated solutions can differ; Second, TOPSISGA varies the crowding operator of NSGA-II [2] substituting the crowding distances with the relative distances D_j^+. We assume two populations: P^t, which represents the current population (size M) during generation t; and P_A^t, which consists of non-dominated solutions (archive size N). Initially, M individuals are randomly generated and the archive of non-dominated solutions is set empty. At each generation t, a combined population $R^t = P^t + P_A^t$ (size M+N) is formed (since all previous population members are included in R^t elitism is ensured). Then R^t is sorted based on dominance (Figs. 2(a) and 2(b)). The following population P_A^{t+1} is established with the non-dominated solutions of R^t starting with the set (F_1) of rank 1 followed by the set (F_2) with rank 2 until the last set (F_L) of rank L. If the count of solutions in all sets from F_1 to F_L is larger than the new population P_A^{t+1} (size N), we sort the last set F_L using the Similarity (D_j^+ in ascending order) (Figs. 2(a) and 2(c)). Afterwards a reproductive selection of individuals randomly selected from P_A^{t+1} is accomplished using a binary tournament and a mating pool (MP) is filled up, at this stage M new individuals are generated by applying recombination operators on MP.

Notice, that the selection operator uses a binary tournament and the criterion is: (1) non-dominated rank -smaller rank (2) similarity -bigger similarity.

4 Experimental Results

4.1 The 0–1 Multiobjective Knapsack Problem

In this section the TOPSISGA method is applied to two problems. The first is the 0–1 multiobjective knapsack problem (0–1 MOKP), which has been widely

studied in the multiobjective community. The second application example is a real world engineering problem in the domain of reliability. Independently of the test problem the R^t population (size $M+N$) was set to 200 individuals, the crossover probability to 0.8, the mutation rate to 0.01, the p value to 2 and the archive size of non-dominated solutions was changed progressively following the sequence N=10, 20, 30, 40, 50, 100 individuals. The maximum generation number (G) for the 0–1 MOKP problem was $G = 500$ and for the reliability problem was $G = 50$, $G = 100$ and $G = 250$.

4.1.1 Description

The 0–1 MOKP problem is well known and has been the subject of in-depth studies in the multiobjective domain. It is easy to implement it, but because of its NP-hard nature, it becomes a very difficult problem to be solved in practice. The 0–1 MOKP can be used to model many real problems and it possesses a high number of applications in finance particularly. Various evolutionary algorithms have been used to solve the 0–1 MOKP, e.g. [7, 8].

The 0–1 MOKP consists of to find a subset of items (weights and profits are associated to each item) maximizing a multiobjective function -expressed as a function of the profit values- and considering the constraints of capacity of each knapsack (maximum weight). The 0–1 MOKP can be defined formally by (1):

$$\begin{cases} max. & f_i(x) = \sum_{j=1}^{m} c_{ij} x_j \qquad i = 1, 2, ..., n \\ Such\ that & \sum_{j=1}^{m} w_{ij} x_j \leq b_i \qquad x_j \varepsilon \{0, 1\} \end{cases} \tag{1}$$

where:
m = number of items
x_j = a decision variable
n = number of objectives
c_{ij} = profit of item j according to knapsack i
w_{ij} = weight of item j according to knapsack i
b_i = capacity of knapsack i
The data adopted have been: two objectives and 100 items, the true Pareto frontier is known (Fig. 3), for more details see:
http://www.tik.ee.ethz.ch/%7ezitzler/testdata.html#testproblems.

4.1.2 Results

Figure 4 shows the results with TOPSISGA and NSGAII (for $G = 500$ and $N = 10$). The labels correspond to the TOPSIS classification of the final front when the weights are $w_1 = w_2 = 0.5$. Notice also that, TOPSISGA focuses upon a particular region of the efficient frontier while NSGA-II finds an even final set.

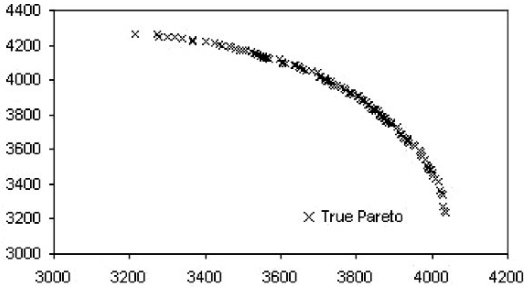


Fig. 3 True Pareto frontier knapsack problem

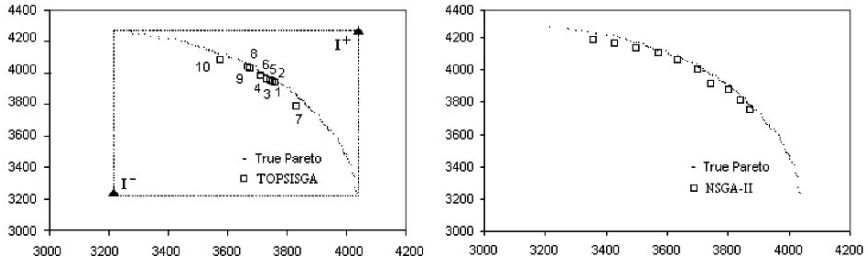


Fig. 4 Non-dominated front found by TOPSISGA and NSGAII

Table 1 Metric C(R,A) values for 500 generations

Method	N = 10	N = 20	N = 30	N = 40	N = 50	N = 100
TOPSISGA	74	74.5	86.27	87.65	84.5	90.95
NSGA-II	99	86	86.33	83.97	87.68	97.83

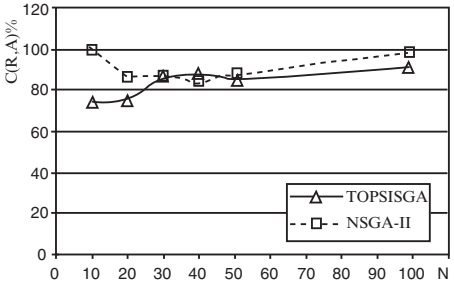


Fig. 5 Graphic view of Table 1

TOPSISGA was compared with NSGA-II based on the C metric [10] (the lower the better), using the efficient frontier as a reference set R. Table 1 reports the percentage (average after ten runs) of the final outcomes (labelled A) dominated by the true Pareto frontier. Figure 5 shows graphically the results of Table 1.

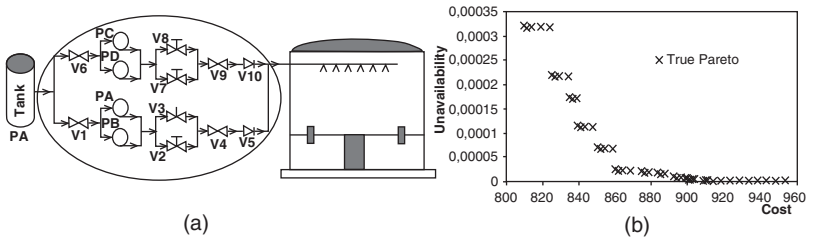


Fig. 6 (**a**) CSIS system for NPP, (**b**) The true Pareto frontier SS problem

Table 2 Component models available on the market

Model	Valves 1,4,6,9	Valves 2,3,7,8	Valves 5,10	Pumps a,b,c,d
Mod. 1	U=2,9E-03 C=50	U=3,0E-03 C=65	U=5,0E-04 C=37	U=3,5E-03 C=90
Mod. 2	U=8,7E-03 U=35	U=1,0E-03 C=70	U=6,0E-04 C=35	U=3,8E-03 C=85
Mod. 3	U=4,0E-04 C=60			

4.2 Safety Systems Design Optimisation

4.2.1 Description

As an application example we will use here a well known dependability problem, the design optimization of a Safety System (SS) -as practical test we use the Containment Spray Injection System (CSIS) of a Nuclear Power Plant (NPP). The problem is combinatorial in nature and NP-hard and it has been widely studied before [4, 5]. In Fig. 6(a) the CSIS layout design is depicted and Table 2 shows the Unavailability and Cost of the different market available components (valves and pumps) for the system, being the optimization purpose to obtain the best design. For each combination of pumps and valves the system unavailability and the system cost are computed, the former using a fault tree with design alternatives and the later using a single aggregating formula. Both objectives are in conflict so a multiobjective optimisation is the appropriate methodology. When any number of objective function evaluations can be made during the optimization, the true Pareto front of non dominated solutions can be obtained using ad hoc multiobjective methods like the NSGA-II [2]. Figure 6(b) shows the true Pareto frontier [5].

4.2.2 Results

Figures 7(a) and 7(b) show the results when the unavailability and the cost are weighted 0.5, 0.5 and 0.8, 0.2 respectively (for G=100 and N=10). The labels correspond to the TOPSIS classification of the final front.

Notice that the final ordering changes with the weights as expected, providing the DM with a final pre-order according to their preferences. Notice also that, while NSGA-II finds an even final set (Fig. 8 for G=100 and N=10), the TOPSISGA focuses upon a particular region of the efficient frontier (Figs. 7(a) and 7(b)) and the final result is far different from the one reached by NSGA-II despite of the fact that the objectives were equally weighted. On the other hand, it is evident that it is impossible to obtain a similar classification to the one obtained by TOPSISGA from the final set presented by NSGA-II and vice versa. It raises the question of where is the right moment to introduce preferences and under what criterion?

Finally, the proposed approach was compared with NSGA-II based on the C metric [10], using the efficient frontier as a reference set R. Table 3 reports the percentage (average after ten runs) of the final outcomes (labelled A) dominated by the true Pareto frontier. Figures 9(a) and 9(b) show graphically the results of Table 3 for 50 and 100 generations.

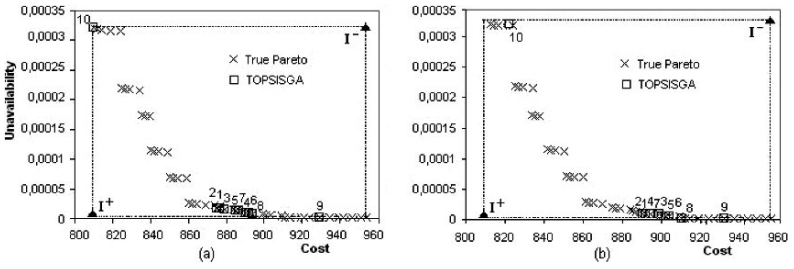


Fig. 7 Non-dominated front found by TOPSISGA

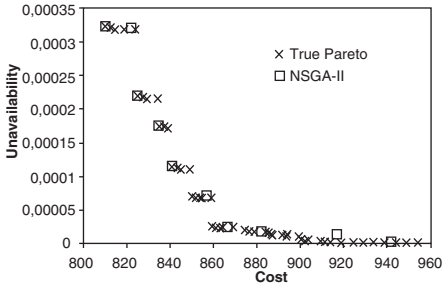


Fig. 8 Non-dominated frontier found by NSGA-II

Table 3 Metric C(R,A) values for 50, 100 and 250 generations

Method	G	N=10	N=20	N=30	N=40	N=50	N=100
TOPSISGA	50	21	10	10.33	8.25	6.38	8.15
NSGA-II		46	29.5	10.33	7.25	7.93	9.45
TOPSISGA	100	18	9.5	9.66	6.5	6.97	4.58
NSGA-II		47	28.55	9.66	4.25	6.96	4.97
TOPSISGA	250	18	6	7	4.5	3.31	2.7
NSGA-II		34	23	4.7	4	4.6	2.47

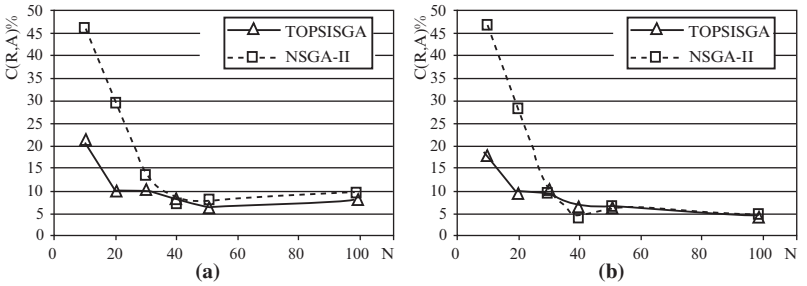


Fig. 9 Graphic view of Table 3: (**a**) 50G, (**b**) 100G

5 Conclusions

In many real discrete problems the number of solutions that belong to the Pareto front is unknown. If the specified size of the non-dominated population in the MOEA is less than the number of solutions of the problem, the found front will be incomplete for a posteriori making decision. In this work we introduce the MOEA structure TOPSISGA that combines the second generation of MOEAs (we use NSGA-II) with the multiple criteria decision making technique TOPSIS. The conducted experiments show that the proportion of efficient frontier reached by the algorithms is larger using TOPSISGA when the archive size of non-dominated solutions is small, but this difference seems to disappear when the archive size of non-dominated solutions increases. Besides, TOPSISGA focuses the search on the region of interest, giving an order list of alternatives in accordance to the DM's preferences. Nevertheless, it could be convenient to find a balance between the spread over the whole front produced by NSGA-II with the identification and the exploitation of the zone of interest realized by TOPSISGA. Kwangsun Yoon [9] measures the credibility of d_p distance function and obtains: the distance function becomes less specific or less credible as parameter p increases. He recommends the use of d_1 for obtaining the most credible compromise solution from the purely mathematical viewpoint. In TOPSISGA we use the p=2 metric, its influences hasn't been checked so far, it is left for future research.

References

1. Cvetkovic D, Coello C (2005) Human preferences and their applications in evolutionary multiobjective optimization. In: Yaochu J (ed) Knowledge incorporation in evolutionary computation (Studies in Fuzziness and Soft Computing, vol. 167). Springer, Berlin, pp 479–502
2. Deb K, Pratap A, Agrawal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
3. Deb K, Sundar J, Udaya Bhaskara Rao N, Chaudhuri S (2006) Reference Point based multiobjective optimization using evolutionary algorithms. Int J Comput Intell Res 2(3):273–286
4. Galván B (1999) Contributions to fault tree quantitative evaluation. Dissertation, Physics Department, Las Palmas de Gran Canaria University, Canary Islands, Spain, (In Spanish)
5. Greiner D, Galván B, Winter G (2003) Safety systems optimum design by multicriteria evolutionary algorithms. In: Evolutionary multi-criterion optimization, LNCS 2632. Springer, Berlin, pp 722–736
6. Hwang CL, Yoon K (1981) Multiple attribute decision making: methods and applications. Springer, Heidelberg
7. Ishibuchi H, Narukawa K (2004) Performance evaluation of simple multiobjective genetic local search algorithms on multiobjective 0/1 knapsack problems. Proc. 2004 Congress on Evolutionary Computation (CEC'2004), IEEE Service Center, vol. 1, pp 441–448
8. Jaszkiewicz A (2002) On the Performance of Multiple Objective Genetic Local Search on the 0/1 Knapsack Problem. A comparative experiment. IEEE Trans Evol Comput 6(4):402–412
9. Kwangsun Yoon (1987) A reconciliation among discrete compromise solutions. J Oper Res Soc 38(3):277–286
10. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput J 8(2):125–148

Multi-Criteria Simple Games

Luisa Monroy and Francisco R. Fernández

Abstract In this paper multi-criteria simple games are introduced. These games constitute an extension of the basic framework of voting systems and related social-choice situations and are a natural tool for modelling these kinds of problems. After introducing and formally defining these games, the special class of monotonic multi-criteria games is characterized. In addition, we analyze core solution concepts for multi-criteria simple games*.

Keywords: Game theory · Multi-criteria analysis · Simple games

1 Introduction

The theory of multi-criteria games analyzes group decision problems when the decision makers consider several criteria, each of which depends on the decision of all players. Studies of games dealing with a multiplicity of criteria have been focused on both, the cooperative and the non-cooperative analysis. For example, see [2–4, 6] and references therein.

In traditional cooperative game theory, a special class of games, the simple games, has been applied to modelling organization and group decision processes, especially when the problems have qualitative outcomes. These games are characterized by the property that each coalition is assigned a value of either 1 (winning coalition) or 0 (losing coalition).

L. Monroy (✉)
Departamento de Economía Aplicada III. Universidad de Sevilla.Avda. Ramón y Cajal 1, 41018-Sevilla, Spain
e-mail: lmonroy@us.es

* This research was partially supported by the Spanish Ministry of Science and Technology projects BEC2003-03111 and MTM2007-67433-C02-01.

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

Simple games constitute a good representation of voting systems in which a number of players or voters are required to collectively accept or reject a given single proposal. Frequently, however, it is unreasonable to consider isolated issues. For instance, in many political processes the problem is not to pick one from among a set of alternatives, but to decide how many of a set of motions will be passed. In this paper, we introduce games with multiple qualitative criteria, where each player simultaneously takes into account a set of criteria and the value of every coalition is given by a one-zero vector, whose non-zero coordinates identify the criteria that the coalition verifies. In this setting, the coalitions are grouped into families in accordance with the criteria verified, leading to the multi-criteria simple games. These games form an appropriate abstract context within which to reason on group decision problems, where numeric utility values are either inappropriate or else impossible to derive. For instance, voting systems and related social choice situations may be modelled as multi-criteria simple games.

As an illustration, suppose that the European Parliament has to pronounce simultaneously on issues such as incorporating new members, applying new taxes to foreign countries and eliminating national parliaments. In this situation the value of every coalition is neither one nor zero, but a one-zero 3-tuple. Thus, it does not make any sense to consider winning and losing coalition in the traditional way, and an analysis of this situation as a multi-criteria simple game should be more appropriate.

The paper is organized as follows. In Sect. 2 we introduce the model together with the basic concepts. The special class of monotonic multi-criteria simple games is introduced and characterized in Sect. 3. In Sect. 4 we describe the multi-criteria simple game in canonical form, which permits the simplification of the model considered. In Sect. 5 core solution concepts for the multi-criteria simple game are analyzed. The paper ends with a section devoted to conclusions.

2 Basic Concepts

Let $N = \{1,2,\ldots,n\}$ be the set of players, where every subset S of N is a coalition, $\mathscr{P}(N)$ is the set of all coalitions and we denote the k qualitative criteria by $C_1, C_2, \ldots, C_k$.

Definition 1. A multi-criteria qualitative game is a pair (N,ϕ) where ϕ is the function $\phi : \mathscr{P}(N) \to \{0,1\}^k$, that assigns to every coalition a one-zero k-tuple, whose non-zero coordinates identify the criteria that the coalition verifies. This vector is called the pattern of coalition.

In one-criterion games, coalitions are classified in accordance with their patterns in winning coalitions (value 1) and losing coalitions (value 0). For the multi-criteria case, $U = \{U_1, U_2, \ldots, U_r\}$ is a classification of the $2^n - 1$ coalitions in $\mathscr{P}(N)$, if $\bigcup_{i=1}^{r} U_i = \mathscr{P}(N)$. When the r classes are mutually disjoint sets, the classification is called a partition.

There are two ways to define a classification on $\mathscr{P}(N)$:

(a) Each class U_i is obtained by establishing different conditions on the criteria.
(b) The pattern of each class is used, where the pattern $\mathbf{P}_i$ of a class U_i is the set of the coalition patterns in the class.

In general, the patterns of the classes are not mutually disjoint sets. However, when the classification is a partition they are always disjoint sets.

A multi-criteria simple game is determined when the classification U or the corresponding family of patterns $\mathbf{P}$ are known.

Consider a classification U on $\mathscr{P}(N)$ and the function $\phi : \mathscr{P}(N) \to \{0,1\}^k$.

Definition 2. A multi-criteria simple game is a pair $(N, v_{\phi,U})$ where $v_{\phi,U} : \mathscr{P}(N) \to \mathscr{P}(U)$ is the characteristic function that assigns to each coalition the set of classes to which it belongs, $v_{\phi,U}(S) = \{U_l, S \in U_l, l \in \{1,\ldots,r\}\}$.

Henceforth, multi-criteria simple games will be denoted by (N, v).

Example 1. (Simultaneous multiple voting). Consider a multiple vote of five candidates and 100 voters, where each voter casts a vote with at most three names and a candidate is chosen if he obtains, at least, 51% of the votes. In this voting system, the vote of each voter is $(w_{i1}, w_{i2}, w_{i3}, w_{i4}, w_{i5})$, where $w_{ij} = 1$ if the voter i votes for the candidate j and $w_{ij} = 0$ otherwise. Therefore $\sum_{j=1}^{5} w_{ij} \leq 3$.

In order to describe this voting process as a multi-criteria simple game, five criteria are considered: C_j, $j \in \{1,2,\ldots,5\}$. The criterion C_j is attained if the candidate j obtains 51 or more votes and the function $\phi : \mathscr{P}(N) \to \{0,1\}^5$ assigns a one-zero 5-tuple to each coalition S . The votes obtained by the candidate j from the voters in coalition S are denoted by $n_j(S)$. Thus, if S is a coalition for which only $n_4(S) \geq 51$ and $n_5(S) \geq 51$, then $\phi(S) = (0,0,0,1,1)$.

We propose the following classification:

(a) The class U_1 is the set of coalitions which verify either, at least the first three criteria or, at least the first two and the last criteria. Its pattern is

$$\mathbf{P}_1 = \{(1,1,1,0,0),(1,1,0,0,1),(1,1,1,1,0),$$

$$(1,1,1,0,1),(1,1,0,1,1)(1,1,1,1,1)\}$$

(b) The class U_2 is the set of coalitions which verify, at least, the fourth and fifth criteria. Its pattern is

$$\mathbf{P}_2 = \{(0,0,0,1,1),(0,0,1,1,1),(0,1,0,1,1),(1,0,0,1,1),$$

$$(1,0,1,1,1),(0,1,1,1,1),(1,1,0,1,1),(1,1,1,1,1)\}$$

(c) The class U_3 is the set of coalitions which verify either, at least the second and fourth criteria or, at least the third and fourth criteria. Its pattern is

$$\mathbf{P}_3 = \{(0,1,0,1,0),(0,0,1,1,0),(1,1,0,1,0),(0,1,1,1,0),$$

$$(0,1,0,1,1),(1,0,1,1,0),(0,0,1,1,1),(1,1,1,1,0),$$
$$(1,1,0,1,1),(0,1,1,1,1),(1,0,1,1,1),(1,1,1,1,1)\}$$

(d) The class U_4 is the set of the remaining coalitions. Its pattern is

$$\mathbf{P}_4 = \{(0,0,0,0,0),(1,0,0,0,0),(0,1,0,0,0),(0,0,1,0,0),$$
$$(0,0,0,1,0),(0,0,0,0,1),(1,1,0,0,0),(1,0,1,0,0),$$
$$(1,0,0,1,0),(1,0,0,0,1),(0,1,0,0,1),(0,0,1,0,1)\}$$

Therefore, we have defined a multi-criteria simple game, (N,v), where the characteristic function, v, assigns to each coalition the set of classes to which it belongs. For instance, consider a coalition S of 76 voters where 40 of them vote for candidates 2, 4 and 5; 16 voters vote for candidates 2, 3 and 4; and 20 voters vote for candidates 4 and 5. The pattern of S is $\phi(S) = (0,1,0,1,1)$ and therefore $v(S) = \{U_2, U_3\}$.

When the pattern of a class has a high number of k-tuples, it may be a tedious job to analyze if a coalition is in the class. However, in many cases, the set of patterns can be represented by its minimal or maximal elements and, therefore, it is easier to confirm if a coalition is in the class. A type of game for which this representation holds is the monotonic multi-criteria simple game, that is introduced in the following section.

3 Monotonic Multi-Criteria Simple Games

There is a special class of multi-criteria simple games: the class of monotonic multi-criteria simple games. These games constitute an important tool for modelling voting systems.

Definition 3. A multi-criteria simple game (N,v) with classification $U = \{U_1, U_2, \ldots, U_r\}$ is monotonic if, for each class $U_i \in U, i = 1,\ldots,r$, one of the following two conditions holds:

(a) $\forall S_1, S_2 \in \mathscr{P}(N)$ such that $S_1 \subseteq S_2$ and $S_1 \in U_i$, then $S_2 \in U_i$.
(b) $\forall S_1, S_2 \in \mathscr{P}(N)$ such that $S_1 \subseteq S_2$ and $S_2 \in U_i$, then $S_1 \in U_i$.

The classes in U which verify the first statement are called positive classes, U_i^+, and the classes in U which verify the second statement are called negative classes, U_i^-.

Example 2. Simultaneous multiple voting in Example 1 is a monotonic multi-criteria simple game.

First, we show that condition 1 is verified by class U_2 (analogously for U_1, U_3) and, secondly, that condition 2 is verified by class U_4.

Consider coalition $S_1 \in U_2$ and $S_1 \subseteq S_2$. Since $S_1 \in U_2$, then at least candidates 4 and 5 have obtained 51 or more votes from the voters in S_1. Thus, any super-coalition of S_1, S_2, will collect at least the same number of votes for candidates 4 and 5 as

in S_1, and therefore $S_2 \in U_2$. This class U_2 does not verify the second condition, since the coalition S_2 of 53 voters where 21 of them vote for candidates 3, 4 and 5, and 32 voters vote for candidates 1, 4 and 5, is in U_2. However, the sub-coalition S_1 formed by 21 voters who vote for candidates 3,4 and 5, and 27 voters that vote for candidates 1, 4 and 5 is not in U_2.

Consider the coalition $S_2 \in U_4$. In this case, at most only one candidate has obtained at least 51 or more votes from the voters in S_2, or only candidates 1 and one of the others have obtained at least 51 or more votes from the voters in S_2, or only candidates 2 and 5 have obtained at least 51 or more votes from the voters in S_2, or only candidates 3 and 5 have obtained at least 51 or more votes from the voters in S_2. Obviously, any sub-coalition of S_2, S_1, will collect fewer votes than in S_2 and therefore no candidate will obtain more votes from S_1 than from S_2. Hence $S_1 \in U_4$. This class does not verify the first condition in the above definition, since coalition S_1 of 63 voters where 20 of them vote for candidates 1, 2 and 3, and 43 voters vote for candidates 1, 3 and 5 is in U_4. However, if 10 new voters, who vote for candidates 2 and 4, join the coalition S_1, then candidates 1,3 and 4 obtain more than 51 votes from the super-coalition S_2, and therefore $S_2 \in U_3$.

In this example, U_1, U_2, U_3 are positive classes and U_4 is a negative class.

There exist multi-criteria simple games which are not monotonic multi-criteria simple games as shown in the following example.

Example 3. Consider a multi-criteria simple game with $N = \{1,2,3,4,5,6\}$ and classification $U = \{U_1, U_2, U_3\}$ given by

(a) The class U_1 is the set of coalitions with three players.
(b) The class U_2 is the set of coalitions with at least four players.
(c) The class U_3 is the set of the remaining coalitions.

In this game, any super-coalition of a coalition in U_1 has more than three players, and hence it is not in U_1. Any sub-coalition of a coalition in U_1 has fewer than three players, and hence it is not in U_1. Therefore, the class U_1 is neither a positive class nor a negative class, and the game is not a monotonic multi-criteria game.

From Definition 3 the concepts of winning coalitions and losing coalitions in scalar simple games can be extended to the multi-criteria case as follows:

Definition 4. Consider a monotonic multi-criteria simple game (N, v) with classification $U = \{U_1, U_2, \ldots, U_r\}$. A coalition in a positive class is called a winning coalition and a coalition in a negative class is called a losing coalition. A coalition is an absolute winning coalition when it belongs to all positive classes.

The following result characterizes monotonic multi-criteria simple games. In order to prove this result, the operations of union and intersection of patterns are considered in the set of patterns **P**.

$$\forall x, y \in \{0,1\}^k, \quad x = (x_1, \ldots, x_k), \quad y = (y_1, \ldots, y_k)$$

$$x \vee y = \max(x,y) = (\max(x_1,y_1),\ldots,\max(x_k,y_k))$$
$$x \wedge y = \min(x,y) = (\min(x_1,y_1),\ldots,\min(x_k,y_k))$$

Theorem 1. *A multi-criteria simple game (N,v) is monotonic if and only if*

$$S_1 \subseteq S_2 \text{ and } S_1 \in U_i^+ \Rightarrow \phi(S_1) \leq \phi(S_2)$$

$$S_1 \subseteq S_2 \text{ and } S_2 \in U_i^- \Rightarrow \phi(S_1) \leq \phi(S_2)$$

Proof. ($\Rightarrow$) Let the multi-criteria simple game be a monotonic game.

Assume $S_1 \subseteq S_2$ and $S_1 \in U_i^+$ and hence $S_2 \in U_i^+$. Denote $\phi(S_1) = x$ and $\phi(S_2) = y$. From the natural order in $\{0,1\}^k$, it follows that $x \leq x \vee y$. Since $S_1 \cup S_2 = S_2$ and $\phi(S_1 \cup S_2) = x \vee y$ then $\phi(S_1) = x \leq x \vee y = \phi(S_1 \cup S_2) = \phi(S_2)$ and therefore $\phi(S_1) \leq \phi(S_2)$.

Assume $S_1 \subseteq S_2$ and $S_2 \in U_i^-$ and hence $S_1 \in U_i^-$, then $\phi(S_1) = \phi(S_1 \cap S_2) = x \wedge y \leq y = \phi(S_2)$. Therefore, $\phi(S_1) \leq \phi(S_2)$.

($\Leftarrow$) The fact that U_i^+ (U_i^-) is a positive (negative) class, leads to the result. □

From this theorem it follows that, for monotonic multi-criteria simple games, the set of patterns is a partially ordered set in accordance with the natural order defined in $\{0,1\}^k$. Therefore, the pattern of each class in a multi-criteria simple game can be given by its minimal (for positive class) or maximal elements (for negative class) and hence, a coalition belongs to a positive (negative) class if the coalition pattern is greater than or equal to (less than or equal to) one of those minimal (maximal) elements.

Thus, in a monotonic multi-criteria simple game, $\phi(N) = (1,1,\ldots,1)$ because if any component of $\phi(N)$ is equal to 0, then it is also equal to 0 $\forall S \in \mathscr{P}(N)$. Thus, the corresponding criterion would never be attained and therefore it could be eliminated. Analogously, $\phi(\emptyset) = (0,0,\ldots,0)$.

In a monotonic multi-criteria simple game the notions of minimal winning coalition and maximal losing coalition are established as follows:

Definition 5. A coalition is a minimal winning coalition (maximal losing coalition) of a class if its pattern is a minimal element, (maximal element), of the pattern of the class.

Example 4. The simultaneous multiple voting considered in Example 1 is a monotonic multi-criteria simple game, therefore, its positive classes are given by their minimal patterns and the negative class is given by its maximal patterns which are:
$\mathbf{P}_1^m = \{(1,1,1,0,0),\ (1,1,0,0,1)\}$, $\mathbf{P}_2^m = \{(0,0,0,1,1)\}$,
$\mathbf{P}_3^m = \{(0,1,0,1,0),\ (0,0,1,1,0)\}$, $\mathbf{P}_4^M = \{(1,1,0,0,0),\ (1,0,1,0,0),\ (1,0,0,1,0),$
$(1,0,0,0,1),\ (0,1,0,0,1),\ (0,0,1,0,1)\}$.

Thus, the coalition S of 51 voters who vote for candidates 4 and 5, is a minimal winning coalition of the class U_2 since $\phi(S) = (0,0,0,1,1)$ is a minimal pattern of the class U_2.

We assume that all the classes, except one, are positive. The negative class, $\mathscr{R}$, is called the residual class and will usually be omitted.

4 Multi-Criteria Simple Games in Canonical Form

Given a multi-criteria simple game (N,v) with k criteria and classification $U = \{U_1, U_2, \ldots, U_r\}$, the associated canonical form is established as a multi-criteria simple game (N,ν) with r criteria, where a coalition verifies the *ith*-criterion if the coalition belongs to the class U_i, $i = 1, \ldots, r$. That is to say, the canonical simple game, (N,ν), is derived from a qualitative game with r criteria, $(N,\hat{\phi})$, where $\hat{\phi} : \mathscr{P}(N) \to \{0,1\}^r$ with

$$\hat{\phi}(S) = (\delta_i(S))_{i=1,\ldots,r} \quad \text{and} \quad \begin{cases} \delta_i(S) = 1 \text{ if } S \in U_i \\ \delta_i(S) = 0 \text{ if } S \notin U_i \end{cases}$$

From this definition it follows that the minimal pattern which defines each class U_i, $i = 1, \ldots, r$, in the canonical form is given by the unit vectors $(e_i)_{i=1,\ldots,r}$. Note that the multi-criteria simple game (N,v) and its associated canonical form (N,ν) have the same classification and, furthermore, $v(S) = \nu(S), \forall S \in \mathscr{P}(N)$. However, when a multi-criteria simple game is represented by its associated canonical form, the classes to which a coalition belongs are known and, therefore, the patterns of each class in the game (N,v) are not needed. For this reason, the use of the canonical form of a multi-criteria simple game is very convenient when the model has a high number of criteria.

Example 5. The canonical form for the simultaneous multiple voting game in Example 1 is (N,ν) where $\hat{\phi}(S) \in \{0,1\}^3$. Thus, if $\hat{\phi}(S) = (0,1,0)$ then $S \in U_2$. Therefore, candidates 4 and 5 have been chosen by the voters in S.

It is important to mention that the associated canonical form of a monotonic multi-criteria simple game is a monotonic multi-criteria simple game, since both games have the same classification. In addition, the canonical simple game, (N,ν), induces r component scalar simple games (N,ν_i), $i = 1, \ldots, r$, where a coalition S is a winning coalition in (N,ν_i) if $S \in U_i$ and it is a losing coalition if $S \notin U_i$.

The voting system in the United Nation Security Council(UNSC) has been considered as an illustration of several game models proposed in the literature, (See [1,5]). In the following example we provide a description of this voting process as a multi-criteria simple game, and then give its canonical form.

Example 6. The UNSC has five permanent members and ten nonpermanent members. A motion is passed if it is supported by at least nine members and no permanent member is explicitly opposed.

To model this voting system as a multi-criteria simple game we establish two criteria: C_1 : There are at least nine "yes" votes, and, C_2 : Each permanent member votes "yes" or "abstain". A classification which determines the game, is $U = \{U_1, \mathscr{R}\}$ where U_1 is the set of coalitions which verifies both criteria, and the remaining coalitions are in $\mathscr{R}$. The patterns of each class are $\mathbf{P}_1 = \{(1,1)\}$ and $\mathbf{P}_{\mathscr{R}} = \{(1,0),(0,1),(0,0)\}$. Thus, the winning coalitions are those in class U_1.

The canonical form for this game is (N,ν) where $\hat{\phi}(S) \in \{0,1\}^2$. Therefore, if $\hat{\phi}(S) = (1,0)$ then $S \in U_1$. The 2 component scalar simple games are (N,ν_1) and

(N, v_2). The first one refers to the alternative "yes" and indicates when a motion is passed. The second game refers to the alternative "no " and establishes when a motion is rejected.

5 The Core

In this section we deal with the core concept in multi-criteria simple games. To this end, the preference core and the dominance core proposed by Fernández et al. [4], are applied.

Scalar simple games may be considered as distributors of political power, and the payoffs can be interpreted as the influence of the players in the attainment of the final outcome. A similar interpretation for multi-criteria simples game is presented.

Let (N, v) be a monotonic multi-criteria simple game in canonical form with classification $U = \{U_1, U_2, \ldots, U_r\}$, and function $\hat{\phi} : \mathscr{P}(N) \to \{0,1\}^r$. The value $\hat{\phi}(N) = (1, \ldots, 1)$ indicates that coalition N has the control over the r classes in the classification. Thus, the value $\hat{\phi}(S)$ of a coalition is its strength or importance in the control over the classes, and therefore, the value $\hat{\phi}(i), i \in N$, represents the influence of the player i in each class. The total control $\hat{\phi}(N)$ can be allocated among the players in any way, but no player will accept less worth than the minimum which he can attain by himself. Hence, an imputation in a multi-criteria simple game represents a dimensionless measure of the distribution of the total power in each class, among the players, and the worth given to each player in each class is, at least, as good as the worth he obtains alone.

Definition 6. An imputation for the monotonic multi-criteria simple game is a matrix

$$X = \begin{pmatrix} x_{11} & x_{21} & \ldots & x_{n1} \\ x_{12} & x_{22} & \ldots & x_{n2} \\ \ldots & \ldots & \ldots & \ldots \\ x_{1r} & x_{2r} & \ldots & x_{nr} \end{pmatrix}$$

satisfying $\sum_{i \in N} X^i = \hat{\phi}(N)^t$ and $X^i \geq \hat{\phi}(i)^t \ \forall i \in N$, where the i-th column, X^i, represents the percentage of power of the player i in each criterion, and the j-th row, X_j, represents the percentage of power of each player in the criterion j.[†]

Definition 7. The preference core of a multi-criteria simple game is the set of imputations X which verify $\sum_{i \in S} X^i \geq \hat{\phi}(S)^t \ \forall S \in \mathscr{P}(N)$.

Definition 8. The dominance core of a multi-criteria simple game is the set of imputations X which verify $\sum_{i \in S} X^i \succeq \hat{\phi}(S)^t \ \forall S \in \mathscr{P}(N)$.

In scalar simple games, the core consists of the imputations that award everything to the veto players. If there are no veto players, there is no core. These two well-known conditions, established by Owen [7], are generalized in this paper for the multi-criteria case. First, we define veto players for this kind of game.

[†] For $x, y \in R^k$ we denote $x \geq y \Leftrightarrow x^i \geq y^i$; $x \succeq y \Leftrightarrow \exists j, x^j > y^j$.

Definition 9. A player i is a *veto player* for a class if he belongs to all coalitions in the class. A player i is an *absolute veto player* if he is a veto player for all the classes.

Theorem 2. *The preference core is nonempty if and only if there is at least one veto player for each class.*

Proof. ($\Rightarrow$) Consider an imputation X in the core and suppose that the game has no veto player. Then, for every $p \in N$, $N\backslash\{p\}$ is an absolute winning coalition and therefore $\sum_{i\neq p} X^i = \hat{\phi}(N)^t$, $\forall p \in N$. Thus, $X^p = (0,\ldots,0)^t \; \forall p \in N$ and X cannot be an imputation.

($\Leftarrow$) Suppose that there is at least one veto player for each class. Let S be the set of all them. Consider a matrix X such that $\sum_{i\in S} X^i = (1,\ldots,1)^t$ and $X^i \geq (0,\ldots,0)^t \; \forall i \in S$, $X^i = (0,\ldots,0)^t \; \forall i \notin S$.

Since $S \subseteq N$, it follows that $\sum_{i\in N} X^i \geq \sum_{i\in S} X^i = (1,\ldots,1)^t = \hat{\phi}(N)^t$, that is, $\sum_{i\in N} X^i = \hat{\phi}(N)^t$, and $X^i \geq \hat{\phi}(i)^t \; \forall i \in N$. Therefore X is an imputation.

Consider $T \in \mathscr{P}(N)$. If $T \in U_j, \forall j \in \{1,\ldots,r\}$ then $S \subseteq T$ and therefore $\sum_{i\in T} X^i \geq \sum_{i\in S} X^i = (1,\ldots,1)^t = \hat{\phi}(T)$.

If $T \in U_j, \forall j \in J \subseteq \{1,\ldots,r\}$ then T contains, at least, the veto players of these classes, and therefore $\sum_{i\in T} X^i = \sum_{i\in S\cap T} X^i + \sum_{i\in T\backslash S} X^i \geq \hat{\phi}(T)$.

Thus, X is in the preference core.

Analogously, the following result holds. □

Theorem 3. *The dominance core is nonempty if and only if there is at least one class with one veto player.*

Example 7. Consider a monotonic multi-criteria simple game, (N,v), with four players, $N = \{1,2,3,4\}$, and two positive classes, U_1, U_2, given by their minimal winning coalitions, which are $\{\{1,2\},\{1,3,4\}\}$ and $\{\{2,3\}\}$, respectively.

Player $\{1\}$ is a veto player for the class U_1 and players $\{2\}$ and $\{3\}$ are veto players for U_2, therefore the preference core is nonempty and is given by

$$\left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_1 & \alpha_2 & 0 \end{pmatrix}, \alpha_1 + \alpha_2 = 1, \alpha_1, \alpha_2 \in R^+ \right\}$$

In this example, player $\{1\}$ has control over class 1 and players $\{2\}$ and $\{3\}$ control class 2 in relation with the other players. However, between these last two players there is no dominance relation. Player $\{4\}$ has no influence in the two classes.

Example 8. Consider a monotonic multi-criteria simple game, (N,v), with $N = \{1,2,3,4\}$ and two positive classes, U_1, U_2, given by their minimal winning coalitions, which are $\{\{1,2\},\{1,3,4\}\}$ and $\{\{2,3\},\{1,4\}\}$, respectively.

Player $\{1\}$ is a veto player for class U_1, but class U_2 has no veto player. Thus, the preference core is empty and the dominance core is given by

$$\left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} \backslash \sum_{i=1}^{4} \alpha_i = 1, \forall \alpha_i \in R^+ \right\}$$

In this case, player $\{1\}$ prevails over the others in the first class. However, no player dominates in the second class.

6 Conclusions

Multi-criteria simple games constitute an important tool for modelling voting systems since the voting process is a group decision-making method under multiple qualitative criteria. When choosing a position or a candidate, qualitative multiple criteria, such as personal characteristics and positions on specific issues, appear in the process.

Despite their apparently simple structure, such games permit the formulation and analysis of the widely studied models in classic coalitional game theory literature. The concepts, properties and results obtained are natural extensions of those proposed in the scalar simple games literature. The solution concept of core, proposed for multi-criteria simple games, provides a global valuation of the importance of the players in the game.

References

1. Amer R, Carreras F, Magaña M (1998) Extension of values to games with multiple alternatives. Ann Oper Res 84:63–78
2. Fernández FR, Puerto J (1996) Vector linear programming in zero-sum multicriteria matrix games. J Optim Theory Appl 89:115–127
3. Fernández FR, Monroy L, Puerto J (1998) Multicriteria goal games. J Optim Theory Appl 99:403–421
4. Fernández FR, Hinojosa M, Puerto J (2002) Core solutions in vector-valued games. J Optim Theory Appl 112:331–360
5. Freixas J, Zwicker WS (2003) Weighted voting, abstention, and multiple levels of approval. Soc Choice Welf 21:399–431
6. Hinojosa MA, Mármol AM, Monroy L (2005) Generalized maxmin solutions in multicriteria bargaining. Ann Oper Res 137:243–255
7. Owen G (1995) Game theory. Academic Press, San Diego

Multiobjective Cooperative Games with Restrictions on Coalitions

Tetsuzo Tanino

Abstract In this paper we consider a multiobjective cooperative game with restrictions on coalitions. We define the restricted game of the original multiobjective cooperative game and discuss its properties, namely inheritance of superadditivity and convexity under appropriate combinatorial structures on the feasible coalition system. We also study the core of the restricted game.

Keywords: Cooperative games · Core · Multiobjective optimization · Restrictions on coalitions

1 Introduction

The theory of cooperative games is quite useful in analyzing decision making situations along with multiple decision makers who can form coalitions. In an ordinary cooperative game (transferable utility game), the results of coalitions are described by worths of coalitions, which are real numbers. On the contrary, in a multiobjective cooperative game, the worth of each coalition is measured by multiple criteria, and therefore it is given as a set in a multidimensional real space [3,5]. This set might be obtained by (Pareto) maximization of an admissible set [7]. Some researchers have studied multiobjective cooperative games and discussed solutions, for example the cores, of them [3,5,7].

On the other hand, cooperative games with some restrictions on coalitions have been actively studied recently (for example Bilbao [1] and Slikker and van den Nouweland [6]). In those cases, the set of feasible coalitions is given as a subset of the power set of the whole player set, and a new game called restricted game is

T. Tanino
Osaka University, Suita, Osaka 565-0871, Japan
e-mail: tanino@eei.eng.osaka-u.ac.jp

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

defined. Solutions of the original game under the restriction on coalitions are obtained as solutions, such as the core or the Shapley value, of the restricted game.

In this paper, we consider a multiobjective cooperative game with restrictions on coalitions. We define the restricted game of the original game and discuss its properties, namely inheritance of superadditivity and convexity. We also study the core of the restricted game.

2 Maximum and Minimum of a Set in $\mathbf{R}^p$

In multiobjective optimization we consider sets in the p dimensional objective real space and maxima and/or minima of those sets. In this paper we use the following notations. First we distinguish two symbols of set inclusions: $A \subseteq B$ means that A is a subset of B, and $A \subset B$ implies that A is a proper subset of B. Let $\mathbf{R}^p$ be the p dimensional real space and $\mathbf{R}^p_+$ the nonnegative orthant in $\mathbf{R}^p$, i.e.,

$$\mathbf{R}^p_+ = \{x = (x_1, \ldots, x_p) \in \mathbf{R}^p \mid x_i \geq 0,\ i = 1, \ldots, p\}.$$

We define the sets Y_+, Y_{++}, Y_-, and Y_{--} for a set $Y \subseteq \mathbf{R}^p$ as follows:

$$\begin{aligned} Y_+ &= Y + \mathbf{R}^p_+,\ Y_{++} = Y + (\mathbf{R}^p_+ \setminus \{\mathbf{0}\}) \\ Y_- &= Y - \mathbf{R}^p_+,\ Y_{--} = Y - (\mathbf{R}^p_+ \setminus \{\mathbf{0}\}), \end{aligned}$$

where $\mathbf{0} = (0, \ldots, 0) \in \mathbf{R}^p$. In terms of these notations, we can define the minimum and maximum of a set in $\mathbf{R}^p$ as follows.

Definition 1 *For a set $Y \subseteq \mathbf{R}^p$, the minimum and maximum of Y are defined by*

$$\begin{aligned} \operatorname{Min} Y &= \{y \in Y \mid (Y - y) \cap (-\mathbf{R}^p_+) = \{\mathbf{0}\}\} = Y \setminus Y_{++} \\ \operatorname{Max} Y &= \{y \in Y \mid (Y - y) \cap \mathbf{R}^p_+ = \{\mathbf{0}\}\} = Y \setminus Y_{--}, \end{aligned}$$

respectively.

A particular type of sets in $\mathbf{R}^p$ satisfies the condition that the minimum or the maximum of a set coincides with the set itself.

Definition 2 *A set $Y \subseteq \mathbf{R}^p$ is said to be thin (with respect to $\mathbf{R}^p_+$) if one of the following equivalent conditions is satisfied:*

$$\begin{aligned} &1)\ Y = \operatorname{Min} Y \\ &2)\ Y = \operatorname{Max} Y \\ &3)\ Y_+ \setminus Y = Y_{++} \\ &4)\ Y_- \setminus Y = Y_{--} \end{aligned}$$

Remark 1 *For any $Y \subseteq \mathbf{R}^p$, the sets* Min Y and Max Y are obviously thin with respect to $\mathbf{R}^p_+$.

3 Multiobjective Cooperative Games

An ordinary cooperative game (transferable utility game) is a pair of a set of players $N = \{1, \ldots, n\}$ and a characteristic function $v : 2^N \to \mathbf{R}$ satisfying $v(\emptyset) = 0$. A subset $S \subseteq N$ is called a coalition and $v(S)$ is the worth of S. In a multiobjective cooperative game this worth should be measured by multiple (p) criteria, and therefore it is specified by a subset of $\mathbf{R}^p$ [3, 5, 7]. Thus a multiobjective cooperative game (MO-game for short) is a pair (N,V), where V is a set-valued mapping from 2^N to $\mathbf{R}^p$, i.e., $V(S) \subseteq \mathbf{R}^p$ for any $S \subseteq N$. We assume that $V(\emptyset) = \{\mathbf{0}\}$ and that $V(S)$ is nonempty, compact and thin for any $S \subseteq N$ throughout this paper. The second condition implies that the multidimensional worth $V(S)$ of S is Pareto efficient in the MO-game. Namely there is no Pareto ordering between two points in $V(S)$. If y is contained in $V(S)_-$, then it should not be contained in $V(S)$.

In practical situations a number of important cooperative games arise from optimization problems (See Curiel [2] for example). Those optimization problems are linear production programming problems, assignment problems, minimum cost spanning tree problems, and so on. They can be extended to multiobjective problems and therefore we can obtain multiobjective cooperative games arising from them. For example, Nishizaki and Sakawa discussed multiobjective linear production programming games in detail [5]. Since solving a multiobjective optimization problem leads to the Pareto efficient set in the objective space, which is regarded as the worth in a multiobjective cooperative game, it is quite natural that this set is thin.

Definition 3 *An MO-game (N,V) is said to be superadditive if*

$$V(S) + V(T) \subseteq V(S \cup T)_-, \ \textit{for all } S, T \subseteq N,\ S \cap T = \emptyset.$$

Remark 2 *From the above definition, if an MO-game (N,V) is superadditive, then for any $S_k \subseteq N$ $(k \in K)$ such that $S_k \cap S_{k'} = \emptyset$ for $k \neq k'$, $\sum_{k \in K} V(S_k) \subseteq V(\bigcup_{k \in K} S_k)_-$.*

Definition 4 *An MO-game (N,V) is said to be convex if*

$$V(S) + V(T) \subseteq [V(S \cup T) + V(S \cap T)]_-, \ \textit{for all } S, T \subseteq N.$$

It is obvious that convexity is a stronger requirement than superadditivity.

4 Restricted Multiobjective Cooperative Games by Partition Systems

In fundamental cooperative games and also in MO-games, it is assumed that an arbitrary subset S of N can form a coalition, i.e., every S is feasible or admissible. In practical situations, however, this assumption is not necessarily valid. Some

coalitions may not be feasible because of physical or ideological reasons. Those situations are dealt with by introducing the concept of feasible coalition system [1]. A set system is a pair $(N,\mathcal{F})$, with $\mathcal{F} \subseteq 2^N$. The sets belonging to $\mathcal{F}$ are called feasible coalitions. For any $S \subseteq N$, maximal feasible subsets of S are called components of S. In many cases we impose appropriate combinatorial structures on $(N,\mathcal{F})$.

Definition 5 *[1] A partition system is a set system satisfying*

(a) $\emptyset \in \mathcal{F}$, *and* $\{i\} \in \mathcal{F}$ *for every* $i \in N$, *and*
(b) for all $S \subseteq N$, *the components of* S, *denoted by* $\Pi_{\mathcal{F}}(S) = \{T_1, \ldots T_l\}$ *form a partition of* S.

Proposition 1 *[1] A set system* $(N,\mathcal{F})$ *which satisfies the first condition of the above definition is a partition system if and only if* $S,T \in \mathcal{F}$ *and* $S \cap T \neq \emptyset$ *imply* $S \cup T \in \mathcal{F}$.

A typical example of a partition system is the communication structure by Myerson [4].

Definition 6 *Let* (N,V) *be an MO-game and let* $(N,\mathcal{F})$ *be a partition system. The* $\mathcal{F}$*-restricted game* $(N,V^{\mathcal{F}})$, *is defined by*

$$V^{\mathcal{F}}(S) = \mathrm{Max} \sum_{T \in \Pi_{\mathcal{F}}(S)} V(T),$$

where $\Pi_{\mathcal{F}}(S)$ *is the collection of the components of* $S \subseteq N$.

Remark 3 *Since* $V(T)$ *is compact for any* $T \subseteq N$, $V^{\mathcal{F}}(S)$ *is also compact and thin. If* $S \in \mathcal{F}$, *then* $\Pi_{\mathcal{F}}(S) = \{S\}$ *and hence* $V^{\mathcal{F}}(S) = V(S)$.

Lemma 1. *Let* $(N,\mathcal{F})$ *be a partition system,* $S,T \subseteq N$ *with* $S \cap T = \emptyset$,

$$\Pi_{\mathcal{F}}(S) = \{S_k\}_{k\in K},\ \Pi_{\mathcal{F}}(T) = \{T_l\}_{l\in L},\ \text{and}\ \Pi_{\mathcal{F}}(S \cup T) = \{U_m\}_{m\in M}.$$

Then $\{S_k\}_{k\in K} \cup \{T_l\}_{l\in L}$ *is a subpartition of* $\{U_m\}_{m\in M}$.

Proof. It is obvious that $\{S_k\}_{k\in K} \cup \{T_l\}_{l\in L}$ is a partition of $\bigcup_{m\in M} U_m = S \cup T$. For each S_k there exists some U_m such that $S_k \cap U_m \neq \emptyset$. Then we can prove that $S_k \subseteq U_m$. In fact, otherwise, $U_m \subset S_k \cup U_m \in \mathcal{F}$ since $\mathcal{F}$ is a partition system, which is a contradiction. Analogously each T_l is contained in a unique U_m. Hence $\{S_k\}_{k\in K} \cup \{T_l\}_{l\in L}$ is a subpartition of $\{U_m\}_{m\in M}$. □

Due to this lemma we can prove the following theorem which shows the inheritance of superadditivity of the original game to the $\mathcal{F}$-restricted game.

Theorem 1 *Let* (N,V) *be a superadditive MO-game and* $(N,\mathcal{F})$ *be a partition system. Then the* $\mathcal{F}$*-restricted game* $(N,V^{\mathcal{F}})$ *is also superadditive.*

Proof. Let $\Pi_{\mathscr{F}}(S) = \{S_k\}_{k\in K}$, $\Pi_{\mathscr{F}}(T) = \{T_l\}_{l\in L}$, and $\Pi_{\mathscr{F}}(S\cup T) = \{U_m\}_{m\in M}$. Then due to Lemma 1 and Remark 1,

$$\begin{aligned} V^{\mathscr{F}}(S)+V^{\mathscr{F}}(T) &= \operatorname{Max}\sum_{k\in K} V(S_k)+\operatorname{Max}\sum_{l\in L} V(T_l) \\ &\subseteq \sum_{k\in K} V(S_k)+\sum_{l\in L} V(T_l) \\ &\subseteq \sum_{m\in M} V(U_m)_- \\ &= [\sum_{m\in M} V(U_m)]_- \\ &= [\operatorname{Max}\sum_{m\in M} V(U_m)]_- \\ &= V^{\mathscr{F}}(S\cup T)_- \end{aligned}$$

The second last equality follows since $\sum_{m\in M} V(U_m)$ is compact. This completes the proof. □

5 Inheritance of Convexity

In this section we consider a more special type of feasible coalition systems called intersecting systems, and prove the inheritance of convexity to the restricted games by intersecting systems.

Definition 7 *A partition system $(N,\mathscr{F})$ is called an intersecting system if for all $S,T\in\mathscr{F}$ with $S\cap T\neq\emptyset$ we have $S\cap T\in\mathscr{F}$.*

Remark 4 *In Bilbao [1], a set system $(N,\mathscr{F})$ is called an intersecting family if for all $S,T\in\mathscr{F}$ with $S\cap T\neq\emptyset$ we have $S\cap T\in\mathscr{F}$ and $S\cup T\in\mathscr{F}$. Therefore an intersecting system is an intersecting family satisfying the first condition, $\emptyset\in\mathscr{F}$ and $\{i\}\in\mathscr{F}$, of the partition system.*

Theorem 2 *Let (N,V) be a convex MO-game and $(N,\mathscr{F})$ be an intersecting system. Then the restricted game $(N,V^{\mathscr{F}})$ is also convex.*

Proof. Let $S,T\subseteq N$. If $S\cap T=\emptyset$, convexity reduces to superadditivity and therefore holds obviously. Hence we assume that $S\cap T\neq\emptyset$ in the proof. Let

$$\Pi_{\mathscr{F}}(S)=\{S_1,\ldots,S_l\},\ \text{and}\ \Pi_{\mathscr{F}}(T)=\{T_1,\ldots,T_m\}.$$

We prove the theorem by induction both in l and in m.

(a) First let $l=1$, i.e., suppose that $\Pi_{\mathscr{F}}(S)=\{S\}$ and therefore $V^{\mathscr{F}}(S)=V(S)$. We prove the relation

$$V(S)+V(T)\subseteq[V(S\cup T)+V(S\cap T)]_-$$

by induction with respect to m. Thus we first consider the case $m = 1$. Then $\Pi_{\mathscr{F}}(T) = \{T\}$ and $V^{\mathscr{F}}(T) = V(T)$. Since $S, T \in \mathscr{F}$ and $(N, \mathscr{F})$ is an intersecting system, $S \cup T, S \cap T \in \mathscr{F}$. Then

$$\begin{aligned} V^{\mathscr{F}}(S) + V^{\mathscr{F}}(T) &= V(S) + V(T) \\ &\subseteq [V(S \cup T) + V(S \cap T)]_- \\ &= [V^{\mathscr{F}}(S \cup T) + V^{\mathscr{F}}(S \cap T)]_-, \end{aligned}$$

since (N, V) is convex. Thus we have proved the case $m = 1$. Next suppose that the result holds for $m = 1, \ldots, k-1$ $(l = 1)$ and prove the case $m = k$. Let $\Pi_{\mathscr{F}}(T) = \{T_1, \ldots, T_k\}$ and we assume without loss of generality that $S \cap T_k \neq \emptyset$. Thus $S \cup T_k, S \cap T_k \in \mathscr{F}$. Let $T' = T_1 \cup \ldots \cup T_{k-1}$. Then $\Pi_{\mathscr{F}}(T') = \{T_1, \ldots, T_{k-1}\}$.

$$\begin{aligned} V^{\mathscr{F}}(S) + V^{\mathscr{F}}(T) &= V^{\mathscr{F}}(S) + \mathrm{Max} \sum_{m=1}^{k} V(T_m) \\ &\subseteq V^{\mathscr{F}}(S) + \sum_{m=1}^{k} V(T_m) \\ &= V^{\mathscr{F}}(S) + V^{\mathscr{F}}(T_k) + \sum_{m=1}^{k-1} V(T_m) \\ &\subseteq V^{\mathscr{F}}(S \cup T_k) + V^{\mathscr{F}}(S \cap T_k) + V^{\mathscr{F}}(T') - \mathbf{R}^p_+ \\ &\subseteq V^{\mathscr{F}}(S \cup T_k \cup T') + V^{\mathscr{F}}((S \cup T_k) \cap T') + V^{\mathscr{F}}(S \cap T_k) - \mathbf{R}^p_+ \\ &= V^{\mathscr{F}}(S \cup T) + V^{\mathscr{F}}(S \cap T') + V^{\mathscr{F}}(S \cap T_k) - \mathbf{R}^p_+ \\ &\subseteq V^{\mathscr{F}}(S \cup T) + V^{\mathscr{F}}((S \cap T') \cup (S \cap T_k)) - \mathbf{R}^p_+ \\ &= [V^{\mathscr{F}}(S \cup T) + V^{\mathscr{F}}(S \cap T)]_-. \end{aligned}$$

Thus the theorem is proved for $l = 1$ and $m = 1, 2, \ldots$.

(b) Now we suppose that the result is valid for $l = 1, \ldots, k-1$ and $m = 1, 2, \ldots$ and prove the case $l = k$ and m is arbitrary. In this case $\Pi_{\mathscr{F}}(S) = \{S_1, \ldots, S_k\}$. We assume without loss of generality that $S_k \cap T \neq \emptyset$ and let $S' = S_1 \cup \ldots \cup S_{k-1}$. Then

$$\begin{aligned} V^{\mathscr{F}}(S) + V^{\mathscr{F}}(T) &= \mathrm{Max} \sum_{l=1}^{k} V(S_k) + V^{\mathscr{F}}(T) \\ &\subseteq \sum_{l=1}^{k} V(S_k) + V^{\mathscr{F}}(T) \\ &= \sum_{l=1}^{k-1} V(S_l) + V(S_k) + V^{\mathscr{F}}(T) \\ &\subseteq V^{\mathscr{F}}(S') + V^{\mathscr{F}}(S_k) + V^{\mathscr{F}}(T) - \mathbf{R}^p_+ \\ &\subseteq V^{\mathscr{F}}(S') + V^{\mathscr{F}}(S_k \cup T) + V^{\mathscr{F}}(S_k \cap T) - \mathbf{R}^p_+ \\ &\subseteq V^{\mathscr{F}}(S' \cup S_k \cup T) + V^{\mathscr{F}}(S' \cap (S_k \cup T)) + V^{\mathscr{F}}(S_k \cap T) - \mathbf{R}^p_+ \end{aligned}$$

$$
\begin{aligned}
&= V^{\mathscr{F}}(S\cup T)+V^{\mathscr{F}}(S'\cap T)+V^{\mathscr{F}}(S_k\cap T)-\mathbf{R}^p_+\\
&\subseteq V^{\mathscr{F}}(S\cup T)+V^{\mathscr{F}}((S'\cap T)\cup(S_k\cap T))-\mathbf{R}^p_+\\
&= [V^{\mathscr{F}}(S\cup T)+V^{\mathscr{F}}(S\cap T)]_-.
\end{aligned}
$$

This completes the proof of the theorem. □

6 The Core of Restricted Games

In a cooperative game, an allocation scheme of the profit among the players is regarded as a solution of the game. For an MO-game, this allocation is described by an np dimensional vector $x=(x^1,\ldots,x^n)$, where each x^i $(i=1,\ldots,n)$ is a p dimensional vector representing a payoff vector received by player i.

The core is a fundamental solution concept not only in cooperative games, but also in MO-games [3, 5, 7]. It is characterized by two types of requirements: group rationality and coalition rationality.

Definition 8 *The core of an MO-game (N,V) is defined by*

$$
C(V)=\left\{x\in\mathbf{R}^{np}\mid \sum_{i\in N}x^i\in V(N),\ \sum_{i\in S}x^i\in V(S)_+ \text{ for all } S\subseteq N\right\}.
$$

Theorem 3 *Let (N,V) be an MO-game and let $(N,\mathscr{F})$ be a partition system such that $V(N)=V^{\mathscr{F}}(N)$, which is true when $N\in\mathscr{F}$. Then*

$$
C(V^{\mathscr{F}})\subseteq\left\{x\in\mathbf{R}^{np}\mid \sum_{i\in N}x^i\in V(N),\sum_{i\in S}x^i\in V(S)_+ \text{ for all } S\in\mathscr{F}\right\}
$$

Moreover, if $\sum_{T\in\Pi_{\mathscr{F}}(S)}V(T)$ is thin for any $S\subseteq N$, then the equality hods in the above relation, and therefore $C(V)\subseteq C(V^{\mathscr{F}})$.

Proof. First let $x\in C(V^{\mathscr{F}})$. Then $\sum_{i\in N}x^i\in V^{\mathscr{F}}(N)=V(N)$ and $\sum_{i\in S}x^i\in V^{\mathscr{F}}(S)_+=V(S)_+$ for any $S\in\mathscr{F}$. Conversely, suppose that $\sum_{i\in N}x^i\in V(N)=V^{\mathscr{F}}(N)$ and $\sum_{i\in S}x^i\in V(S)_+=V^{\mathscr{F}}(S)_+$ for all $S\in\mathscr{F}$. Take $S\notin\mathscr{F}$ and let $\Pi_{\mathscr{F}}(S)=\{S_k\}_{k\in K}$. Then

$$
\begin{aligned}
\sum_{i\in S}x^i &= \sum_{k\in K}\sum_{i\in S_k}x^i\in\sum_{k\in K}V(S_k)_+\\
&= [\sum_{k\in K}V(S_k)]_+=[\mathrm{Max}\sum_{k\in K}V(S_k)]_+=V^{\mathscr{F}}(S)_+.
\end{aligned}
$$

Hence $x\in C(V^{\mathscr{F}})$, as was to be proved. □

7 Conclusion

We have defined the $\mathcal{F}$-restricted game $(N, V^{\mathcal{F}})$ for a multiobjective cooperative game (N, V) and a partition system $(N, \mathcal{F})$. It is shown that superadditivity is inherited from (N, V) to $(N, V^{\mathcal{F}})$. Inheritance of convexity is guaranteed when $(N, \mathcal{F})$ is an intersecting system. We have also considered the core of $(N, V^{\mathcal{F}})$ and proved that it can be specified by the original game (N, V) under some condition.

References

1. Bilbao JM (2000) Cooperative games on combinatorial structures. Kluwer, Boston
2. Curiel I (1997) Cooperative game theory and applications. Kluwer, Boston
3. Fernández FR, Hinojosa MA, Puerto J (2004) Set-valued TU-games. Eur J Oper Res 159:181–195
4. Myerson R (1977) Graphs and cooperation in games. Math Oper Res 2:225–229
5. Nishizaki I, Sakawa M (2001) Fuzzy and multiobjective games for conflict resolution. Physica, Heidelberg
6. Slikker M, van den Nouweland A (2001) Social and economic networks in cooperative game theory. Kluwer, Boston
7. Tanino T, Muranaka Y, Tanaka M (1992) On multiple criteria characteristic mapping games. Proc MCDM (Taipei) 2:63–72

An Experimental Investigation of the Optimal Selection Problem with Two Decision Makers

Fouad Ben Abdelaziz and Saoussen Krichen

Abstract We explore the optimal selection problem where two decision makers are involved in the evaluation of the arriving offers. We develop three stopping rules to avoid conflictual situations where a decision makers agrees with a current offer and the other wishes to discard it. The three stopping rules are then implemented in C language and runned on a series of problems with various sizes. The experimental results show that the selection rules generate different behaviours of the decision makers. Finally, we compare our problem with the optimal selection problem including a single decision maker and show the loss of utility generated by the introduction of an additional decision maker.

Keywords: Dynamic programming · Game theory · Group decisions · Optimal stopping

1 Introduction

The optimal selection problem (OSP) is widely studied in the literature because of its importance in modelling and solving various practical situations as the selection of projects, the purchase of assets in stock exchanges, and the search of petroleum location. The basic version of the OSP was formulated and solved in [3]. The OSP consists in a sequence of a fixed number of n offers observed by a *single* decision maker (DM) with the objective of maximizing his utility known as the "nothing

S. Krichen (✉)
*College of engineering, American University of Sharjah, PO BOX 26666, Sharjah, UAE; **LARODEC Laboratory, Institut Supérieur de Gestion, University of Tunis, 41 Rue de la Liberté, Le Bardo 2000, University of Tunis, Tunisia
e-mail: saoussen.krichen@isg.rnu.tn

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

but the best" utility. The DM is rewarded by 1 if he chooses the best offer and 0 otherwise. The utility of the DM is expressed as follows:

$$v(k) = \begin{cases} 1 \text{ if } k = 1 \\ 0 \text{ otherwise.} \end{cases} \quad (1)$$

The main dilemma of the OSP is that if an offer is refused and has a relative rank 1, it can be the best of the sequence. However, if it is accepted, there is a nonzero probability to be not the best. In this case, the DM is rewarded by 0. Indeed, if an offer is once observed and refused, it cannot be re-observed in later stages. We investigate in this paper an OSP where *two DMs* are involved in the selection of a single offer. We call this problem the bilateral optimal selection problem (BOSP). n offers are observed one at a time by both DMs. At each stage of the selection process, a group decision should be taken: accept the current offer and stop sampling or discard it and examine the next offer. We assume that no recall of previously examined offers is allowed: if an offer is once discarded, it cannot be re-examined in later stages. A conflict arises when one DM decides to accept a currently inspected offer and the second decides to discard it. In such conflicting situations a decision should be taken by defining a stopping rule for the group. We propose to explore three stopping rules namely:

The selection process is stopped if	• *Either DM accepts an offer* • *Both DMs accept an offer* • *An interactive dynamic approach of the two individual decisions yields to an acceptance*

We develop for each of the above stopping rules its dynamic programming formulation. Then, illustate and detail the above approaches by an experimental investigation using a C++ computer code for various sizes of the problem. We compare the bilateral situation to the single DM selection problem by providing expected utilities of the DM for the same sizes of problems for the BOSP.

2 Dynamic Programming Formulation

We develop in this section the backward recursive equations related to each stopping rule. For all cases, the decision is taken in terms of two main components: the expected utility when stopping and the expected utility when continuing the selection process. These two expressions are different from case to case in the sense that the induvidual decision of the opponent is integrated differently when changing the stopping rule. In what follows, we develop the dynamic equations when each DM_p ($p = 1, 2$) tries to select an offer out of the r_p best offers (r_p is a prefixed threshold by DM_p). So, each DM's utility is:

$$v^p(k_p) = \begin{cases} 1 \text{ if } k_p \leq r_p \\ 0 \text{ otherwise.} \end{cases} \quad (2)$$

2.1 A BOSP Allowing a Stop by One DM

Since the utility of each DM is expressed in terms of the absolute rank of the selected offer and only relative ranks are available during the selection process, the conditional probability that the current offer has an absolute rank k such that its relative rank at stage i is k_p should be outlined. The corresponding probability (3) that the ith offer with relative rank k_p has an absolute rank k is expressed as follows [6]:

$$P\{(n,k)|(i,k_p)\} = \frac{C_{k-1}^{k_p-1} C_{n-k}^{i-k_p}}{C_n^i}. \quad (3)$$

The optimality equation of DM_p $(p = 1,2)$ at stage i when the relative rank of the ith offer is k_p for DM_p and $k_{p'}$ for $DM_{p'}$ $(p, p' \in \{1,2\})$ is:

$$Ev^p(i,k_p,k_{p'}) = \begin{cases} max\{Ev_s^p(i,k_p), Ev_c^p(i)\} & \text{if } i = 1,\ldots,n-1 \\ v^p(k_p) & \text{if } i = n. \end{cases} \quad (4)$$

For instance, when developing the optimality equations of each DM_p $(p = 1,2)$, the stopping part $Ev_s^p(i,k_p)$ at stage i is computed regardless the opponent's individual decision. So, DM_p behaves when deciding to stop as in the single DM framework. Accordingly, DM_p's expected utility if he decides to stop sampling by accepting the ith offer with relative rank k_p is:

$$Ev_s^p(i,k_p) = \sum_{k=k_p}^{r_p} \frac{C_{k-1}^{k_p-1} C_{n-k}^{i-k_p}}{C_n^i}. \quad (5)$$

The expected utility of DM_p $(p = 1,2)$ when he decides to discard the ith offer by following the optimal strategy is:

$$Ev_c^p(i,k_{p'}) = \frac{1}{(i+1)^2} \sum_{j=1}^{i+1} \sum_{k=1}^{i+1} Ev^p(i+1,j,k)\chi_E, \quad (6)$$

where χ_E denotes the indicator function of the event:

$E = \{Ev_s^{p'}(i,k_{p'}) - \frac{1}{(i+1)^2} \sum_{j=1}^{i+1} \sum_{k=1}^{i+1} Ev^{p'}(i+1,j,k) < 0\}$. That is

$$\chi_E = \begin{cases} 1 \text{ if } E \text{ occurs} \\ 0 \text{ otherwise.} \end{cases} \quad (7)$$

2.2 A BOSP Allowing a Stop by Both DMs

We reconsider in this section the BOSP with an alternative stopping rule for the group: the acceptance of an offer requires the individual acceptance of the two DMs. We develop the backward recursive equations for DMs 1 and 2 regarding the problem assumptions and the utilities in (2).

We remind that the optimality equation for DM_1, at stage i, is written as follows:

$$Ev^1(i,k_1,k_2) = \begin{cases} max\{Ev_s^1(i,k_1,k_2), Ev_c^1(i)\} & \text{if } i = 1,\ldots,n-1 \\ v^1(k_1) & \text{if } i = n. \end{cases} \tag{8}$$

The expected utility for DM_1, conditioned by DM_2's acceptance, if he decides to accept the current offer is:

$$Ev_s^1(i,k_1,k_2) = \sum_{k=k_1}^{r_1} \frac{C_{k-1}^{k_1-1} C_{n-k}^{i-k_1}}{C_n^i} \chi_E. \tag{9}$$

The value $\sum_{k=k_1}^{r_1} \frac{C_{k-1}^{k_1-1} C_{n-k}^{i-k_1}}{C_n^i}$ denotes the expected individual utility of DM_1 when the absolute rank of the selected offer varies from k_1 to r_1.

The expected utility of DM_1 if he decides to continue the selection process is:

$$Ev_c^1(i) = \frac{1}{(i+1)^2} \sum_{k=1}^{i+1} \sum_{l=1}^{i+1} Ev^1(i+1,k,l) \tag{10}$$

DM_2 is allowed to decide about the ith offer only if DM_1 accepts it therefore:

$$Ev^2(i,k_2) = \begin{cases} max\{Ev_s^2(i,k_2), Ev_c^2(i)\} & \text{if } DM_1 \text{ accepts the } i\text{th offer} \\ & \text{and } i = 1,\ldots,n-1 \\ Ev_c^2(i) & \text{if } DM_1 \text{ rejects the } i\text{th offer} \\ & \text{and } i \neq n, \\ v^2(k_2) & \text{if } i = n. \end{cases} \tag{11}$$

DM_2 is allowed to accept or reject the current offer if DM_1 accepts and $k_2 \leq r_2$. In this case, his expected utility if he decides to accept the ith offer is:

$$Ev_s^2(i,k_2) = \sum_{k=k_2}^{r_2} \frac{C_{k-1}^{k_2-1} C_{n-k}^{i-k_2}}{C_n^i} \tag{12}$$

And his expected utility when continuing the process is:

$$Ev_c^2(i) = \frac{1}{(i+1)^2} \sum_{k=1}^{i+1} \sum_{l=1}^{i+1} Ev^2(i+1,k,l). \tag{13}$$

2.3 An Interactive Dynamic Stopping Rule for the BOSP

The interactive approach described in [2] is a stopping rule allowing, at each stage of the selection process, the consideration of the two standpoints, then their aggregation taking into account previous stages. We assume that the utilities of the DMs are described in equation (2). The individual expected utility of DM_p ($p = 1, 2$), with relative rank of the ith offer k_p, when he decides to select the current offer and stop the process is the following:

$$s^p(i,k_p) = \sum_{j=k_p}^{n-i+k_p} P\{(n,j)|(i,k_p)\}v^p(j), \tag{14}$$

Where

$$P\{(n,j)|(i,k_p)\} = \frac{C_{k-1}^{k_p-1} C_{n-k}^{i-k_p}}{C_n^i}. \tag{15}$$

The expected utility of DM_p when stopping the process is 0 when the group decision is to continue sampling, and $s^p(i,k_p)$ otherwise. This is expressed as follows:

$$Ev_s^p(i,k_p) = \begin{cases} s^p(i,k_p) & \text{if } \lambda_i^1 s^1(i,k_1) + \lambda_i^2 s^2(i,k_2) \geq \lambda_i^1 Ev_c^1(i) + \lambda_i^2 Ev_c^2(i) \\ 0 & \text{elsewhere.} \end{cases} \tag{16}$$

The expected utility of DM_p at the ith stage if he decides to discard the ith offer and continue according to his individual optimal strategy is:

$$Ev_c^p(i) = \frac{1}{i+1} \sum_{j=1}^{i+1} Ev^p(i+1,j). \tag{17}$$

The group decision is taken by aggregating the individual expected utilities of the DMs using weights. The following algorithm describes main steps of the interactive approach.

Where

$$\begin{aligned} \lambda_{i+1}^1 &= min\{\lambda_i^1 + \tfrac{Ev_s^1(i,k_1) - Ev_c^1(i)}{Ev_s^1(i,k_1)}, 1\} \\ \lambda_{i+1}^2 &= 1 - \lambda_{i+1}^1. \end{aligned} \tag{18}$$

3 Experimental Results

The above dynamic equations were implemented for each stopping rule in C++ language and experiments were conducted on a series of BOSPs with sizes varying from $n = 5$ to $n = 300$. Some interpretations are noticed when observing the expected utiltities of each DM at stage 1. They correspond to the winning probabilities

Table 1 The interactive algorithm for the BOSP

Initialization:

- *Compute individual optimal strategies for DMs 1 and 2.*
- *Ask the two players to jointly propose initial aggregating weights* λ_0^1 *and* λ_0^2. *If not let* $\lambda_0^1 = \lambda_0^2 = \frac{1}{2}$.

Iterative process:

At any stage i ($i = 1, \ldots, n$) *one of the following cases holds:*

- $i = n$, *then STOP and accept the nth offer.*
- *DMs 1 and 2 accept the ith offer, then offer i is accepted* $\Rightarrow$ *STOP sampling.*
- *DMs 1 and 2 refuse the ith offer, then offer i is refused* $\Rightarrow$ *CONTINUE sampling.*
- *Either DM accepts and the other refuses then the overall decision will be taken in the following way:*

$$\text{If } \lambda_i^1 Ev_s^1(i,k_1) + \lambda_i^2 Ev_s^2(i,k_2) \begin{cases} \geq \lambda_i^1 Ev_c^1(i) + \lambda_i^2 Ev_c^2(i) & \text{STOP with the ith offer as a compromise.} \\ < \lambda_i^1 Ev_c^1(i) + \lambda_i^2 Ev_c^2(i) & \text{CONTINUE sampling. and update the } \lambda\text{-weights as in (18)} \end{cases}$$

of the DM. Other observations related to the expected rank and the comparison of the BOSP with the single DM OSP are also anlayzed.

3.1 Winning Probabilities

Different values of r_p are considered in these experiments:

(a) $r_1 = r_2 = 1$: each DM is interested by his best offer.
(b) $r_1 = r_2 = 3$: In this case each DM tries to select an offer out of the three best offers.
(c) $r_1 = 1$ and $r_2 = 3$: The utilities of the DMs are different.

For each utility and each size of the problem, the expected utility of each DM_p is generated. The numerical results are displayed in Tables 2 and 3.

The first column in each table (2 and 3) enumerates the number of offers considered by the players. The second, the third and the fourth column of each table (2 and 3) displays the probability that DMs 1 and 2 respectively select one out of the r_1 best and one out of the r_2 best of the sequence of n offers.

In what follows, we try to compare the two first stopping rules developed in Sect. 2: stopping if either DM decides to stop, and stop if both DMs decide to accept. Winning probabilities for the case $r_1 = r_2 = 3$ are plotted in Fig. 1. The dashed curve of Fig. 1 correponds to the series of winning probabilities when stopping if either DM decides to stop, and the other line corresponds to the case where an offer is accepted if both DMs decide to stop and accept it. It follows from the above results that under an optimal play of both players:

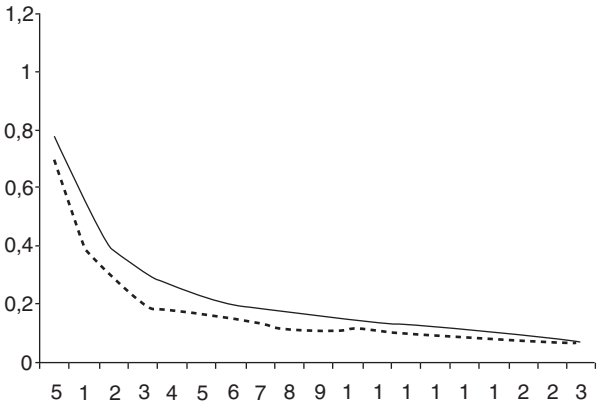


Fig. 1 Winning probabilities for $r_1 = r_2 = 3$ with the two first stopping rules

Table 2 Expected winning probabilities of the DMs if either DM decides to stop the selection process

	$r_1 = r_2 = 1$		$r_1 = r_2 = 3$		$r_1 = 1$ and $r_2 = 3$	
n	DM_1	DM_2	DM_1	DM_2	DM_1	DM_2
5	0.28	0.28	0.691	0.691	0.28	0.713
10	0.21	0.21	0.406	0.406	0.152	0.521
20	0.15	0.15	0.295	0.295	0.115	0.303
30	0.1	0.1	0.212	0.212	0.081	0.0274
40	0.1	0.1	0.166	0.166	0.081	0.261
50	0.1	0.1	0.175	0.175	0.066	0.217
60	0.08	0.08	0.15	0.15	0.056	0.186
70	0.08	0.08	0.131	0.131	0.06	0.192
80	0.075	0.075	0.116	0.116	0.053	0.171
90	0.068	0.068	0.104	0.104	0.057	0.177
100	0.07	0.07	0.117	0.117	0.052	0.162
110	0.06	0.06	0.107	0.107	0.047	0.149
120	0.059	0.059	0.099	0.099	0.051	0.156
130	0.06	0.06	0.092	0.092	0.047	0.145
140	0.057	0.057	0.086	0.086	0.044	0.136
150	0.053	0.053	0.081	0.081	0.041	0.128
200	0.05	0.05	0.074	0.074	0.04	0.122
250	0.044	0.044	0.07	0.07	0.036	0.109
300	0.04	0.04	0.059	0.059	0.03	0.092

- The priority does not provide any advantage for the leader in the game: when $r_1 = r_2$, the players are rewarded by the same utilities.
- When observing the expected utilities in Tables 2 and 3, we can notice that if each player's utility is to select the best offer of the sequence ($r_1 = r_2 = 1$), they are rewarded by the minimum expected values regarding the other cases. So, more the players consider relaxed utilities, more their winning probabilities increase.

Table 3 Expected winning probabilities for the BOSP when stopping if both DMs accept to stop

	$r_1 = r_2 = 1$		$r_1 = r_2 = 3$		$r_1 = 1$ and $r_2 = 3$	
n	DM_1	DM_2	DM_1	DM_2	DM_1	DM_2
5	0.308	0.308	0.775	0.775	0.360	0.745
10	0.191	0.191	0.559	0.559	0.246	0.473
20	0.118	0.118	0.389	0.389	0.165	0.298
30	0.088	0.088	0.31	0.31	0.127	0.231
40	0.070	0.070	0.26	0.26	0.104	0.185
50	0.059	0.059	0.226	0.226	0.09	0.159
60	0.052	0.052	0.201	0.201	0.08	0.14
70	0.046	0.046	0.18	0.18	0.07	0.123
80	0.041	0.041	0.166	0.166	0.064	0.112
90	0.038	0.038	0.154	0.154	0.059	0.103
100	0.035	0.035	0.143	0.143	0.055	0.095
110	0.032	0.032	0.133	0.133	0.051	0.089
120	0.03	0.03	0.126	0.126	0.048	0.084
130	0.028	0.28	0.119	0.119	0.045	0.079
140	0.026	0.026	0.113	0.113	0.042	0.073
150	0.025	0.025	0.107	0.107	0.04	0.069
200	0.02	0.02	0.087	0.087	0.032	0.056
250	0.016	0.016	0.074	0.074	0.027	0.047
300	0.014	0.014	0.064	0.064	0.023	0.04

- If $r_1 \neq r_2$, the players are not rewarded by the same payoffs. We can notice that the player who considers a more relaxed utility ($r_1 = 3$) obtains a better expected utility compared to his opponent.
- We can easily deduce from Fig. 1 that when adopting the second stopping rule (stop if both DMs decide to stop) the winning probability of each DM is greater. So, it is more suitable for the group to adopt the second stopping rule. We can conclude from Tables 2 and 3 that the second stopping rule allows best expected utilities.

3.2 Expected Rank

Since the second stopping rule (stop if both DMs decide to stop) is more suitable for the group, we apply it on a series of problems with $n \in \{5, \ldots, 100\}$. For the stop by both DMs rule, we computed the expected minimum rank at stage 1. This is done by replacing the utility of (2) by $v^p(k_p) = k_p$ and changing all the maximizations by minimizations in the dynamic equations. These expected ranks are dipslayed in the second column of Table 4. For the interactive approach, we generated randomly a sample of 10,000 permutations of the BOSP for each size of the problem. The mean of the obtained ranks is recorded in the third column of Table 4. We notice from Table 4 that for $n = 10$, the expected rank for the second stopping rule and

Table 4 Expected and mean ranks for the second and third stopping rules respectively

n	Both DMs	Negotiation
10	3.99	3.571
20	7.17	5.16
30	10.30	6.40
40	12.70	7.33
50	16.45	8.20
60	19.07	8.975
70	23.14	9.64
80	25.75	10.288
90	28.17	10.888
100	32.33	11.451

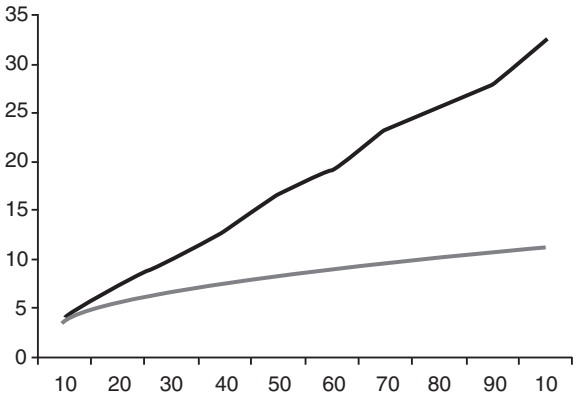


Fig. 2 Expected ranks by second and the third stopping rules

the mean rank for the interactive approach are almost equal. However, for larger n, the mean rank far exceeds the expected rank. This can be clearly seen in Fig. 2. The gray line in Fig. 2 denotes the mean rank by the interacive approach while the black line corresponds to the expected rank when stopping if both DMs decide to stop. We can see in Fig. 2 the gap between the two stopping rules with a sample of offers with various permutations.

Moreover, we can say that for large n a sample of 10,000 permutations against $n!$ possible permutations does not provide an accurate idea about the real mean rank.

3.3 Comparison with the Single DM Case

An alternative investigation of the problem consists in comparing the BOSP and the OSP with a single DM formerly developed by Gusein-Zade [6]. The comparison consists in observing the impact of considering an additional DM in the selection process. Therefore, we consider the BOSP with $r_1 = r_2 \in \{1,2,3\}$ compared with

Table 5 The choice of an offer out of the r best in the single and bilateral DM cases

n	$r = r_1 = r_2 = 1$		$r = r_1 = r_2 = 2$		$r = r_1 = r_2 = 3$	
	$r = 1$	$r_1 = r_2 = 1$	$r = 2$	$r_1 = r_2 = 2$	$r = 3$	$r_1 = r_2 = 3$
5	0.433	0.28	0.7	0.513	0.88	0.691
10	0.398	0.21	0.636	0.287	0.791	0.406
20	0.384	0.15	0.604	0.216	0.747	0.295
30	0.378	0.1	0.594	0.194	0.734	0.212
40	0.375	0.1	0.588	0.152	0.728	0.166
50	0.374	0.1	0.585	0.125	0.723	0.175
60	0.373	0.08	0.583	0.13	0.721	0.15
70	0.372	0.08	0.582	0.114	0.719	0.131
80	0.371	0.075	0.581	0.101	0.718	0.116
90	0.371	0.068	0.58	0.09	0.716	0.104
100	0.371	0.07	0.579	0.098	0.716	0.117
110	0.37	0.06	0.579	0.09	0.715	0.107
120	0.37	0.059	0.578	0.083	0.714	0.099
130	0.37	0.06	0.578	0.077	0.714	0.092
140	0.37	0.057	0.577	0.072	0.713	0.086
150	0.369	0.053	0.577	0.078	0.713	0.081
200	0.369	0.05	0.576	0.06	0.712	0.074
250	0.369	0.044	0.575	0.055	0.711	0.070
300	0.368	0.04	0.575	0.052	0.71	0.059

the OSP where the utility of the DM is 1 if the selected offer is one out of the 3 best offers and 0 otherwise. We implemented the dynamic approach for the single DM case in C++ language and generated the winning probabilities for the DM. Table 5 contains the expected utilities of the DM for the same sizes of the sequences and same thresholds $r = 1, 2, 3$. Table 5 contains the generated results for both single and bilateral cases. In the bilateral case, we provide a single value for each problem since the expected utility at stage 1 is the same for both players. We clearly see how the expected utility of each player decreases compared with the single DM case.

4 Conclusion

We studied in this paper the bilateral optimal selection problem. We proposed three stopping rules namely: either DM can decide to stop, or when both DMs decide to stop, or after an aggregation of the two individual decisions.

We have written the dynamic equations for all the proposed stopping rules. Then, we investigated and compared the stopping rules on a set problems for $n = 5$ to $n = 300$. The stopping rules provide, as expected, different results. Surprisingly, stopping the process by the acceptance of both DMs provide more expected utility than stopping by either of the DMs. This can be explained by the fact that when

either DM can stop the process, he tends to stop earlier. Compared with the case of a single DM, experimental results in Sect. 3 illustrate the loss of expected utility when considering two DMs.

References

1. Ben Abdelaziz F, Krichen S (2007) Optimal stopping by two or more deicision makers: a survey. Comput Manage Sci 4:89–111.
2. Ben Abdelaziz F, Krichen S (2005) An interactive method for the optimal selection problem with two decision makers. Eur J Oper Res 162:602–609
3. Gilbert J, Mosteller F (1966) Recognizing the maximum of a sequence. Am stat Assoc J 61:35–73
4. Krichen S, Ben Abdelaziz F (2007) An optimal stopping problem with two decision makers. Seq Anal 26(4):467–480
5. Szajowski K, Yasuda M (1997) Voting procedure on stopping games of Markov chain. Lect Notes Econ Math Syst 445:68–80
6. Gusein-Zade SM (1966) The problem of choice and the optimal stopping rule for a sequence of independent trials. Theory Probab Appl 11:472–476

Solving a Fuzzy Multiobjective Linear Programming Problem Through the Value and the Ambiguity of Fuzzy Numbers*

Mariano Jiménez, Mar Arenas, Amelia Bilbao, and M^a Victoria Rodríguez

Abstract In this paper we solve multiobjective programming problems with fuzzy parameters and flexible constraints. To work with fuzzy numbers we use two real indices, the value and the ambiguity. In order to rank two fuzzy numbers a lexicographic ranking procedure can be used: in the first step the values of fuzzy numbers are compared, if these values are 'approximately equal' then we compare their ambiguities. In this paper we apply this ranking procedure to a fuzzy programming problem with fuzzy coefficients. In the first step we solve a model in which the fuzzy coefficients have been defuzzified by its corresponding value. Now the question is to determine when two solutions of the first step are approximately equal. In order to answer this question we propose to reflect the decision makers (DMs) preferences through the natural language, establishing a semantic correspondence for the different satisfaction degrees. We consider as approximately equal all the solutions whose global satisfaction degrees belong to the same linguistic label. Then, in the second step, among all the solutions that belong to the same linguistic label as the solution obtained in the first step, we find those with minimum (or maximum) ambiguity, depending on DMs attitude when faced with the risk. We use one example to illustrate this procedure.

Keywords: Fuzzy multi-objective linear programming · Fuzzy numbers · Linguistic label · Value and ambiguity of a fuzzy number

M. Jiménez (✉)
Universidad del País Vasco. Plaza Oñati 1, 20018-San Sebastián, España
e-mail: mariano.jimenez@ehu.es

*Supported by the Universidad del País Vasco (project UPV05/45) and the Spanish Ministerio de Edocación y Ciencia (project MTM2004-07478)

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

1 Introduction

Decision-Makers (DMs) usually take into consideration criteria of a diverse nature in their decision-making process and in general some of them clash and simultaneous optimization of all objectives is impossible. For these reasons many DMs prefer to follow a satisfaction criterion (in order to reach certain aspiration levels) rather than an optimization one. But often these aspiration levels cannot be stated with precision and are expressed in an imprecise way, i.e. "to improve the present profit situation considerably". Furthermore, when addressing real-world problems, frequently the parameters are non-random uncertainness or imprecise numerical quantities. Fuzzy numbers are very adequate for modelling these situations, being that their membership function represents the possibility distribution of uncertain coefficients. Thus in this paper we deal with a flexible programming problem with fuzzy coefficients,

$$\begin{array}{ll} \max & \tilde{c}_r x \qquad r = 1,\ldots,q \\ \text{s.t.} & \tilde{a}_i x \lesssim b_i \; i = 1,\ldots,m \\ & x \geq 0 \end{array} \tag{1}$$

where $\tilde{c}_r x = \tilde{c}_{r1}x_1 \oplus \tilde{c}_{r2}x_2 \oplus \cdots \oplus \tilde{c}_{rm}x_n$, $\tilde{a}_i x = \tilde{a}_{i1}x_1 \oplus \tilde{a}_{12}x_2 \oplus \cdots \oplus \tilde{a}_{in}x_n$, $b_i \in \mathbb{R}$, $x \in \mathbb{R}^n$.

$\tilde{c}_{rj} = (c^1_{rj}, c^2_{rj}, c^3_{rj}, c^4_{rj}), \tilde{a}_{ij} = (a^1_{ij}, a^2_{ij}, a^3_{ij}, a^4_{ij})$ with $c^\bullet_{\bullet\bullet}, a^\bullet_{\bullet\bullet} \in \mathbb{R}$, are fuzzy numbers, $[c^1_{rj}, c^4_{rj}], [a^1_{ij}, a^4_{ij}]$ being their supports and $[c^2_{rj}, c^3_{rj}], [a^2_{ij}, a^3_{ij}]$ their core. $\oplus$ denotes the extended minimum addition [9]. That is to say, given two fuzzy numbers $\tilde{A}, \tilde{B}$, if $\tilde{A} \oplus \tilde{B} = \tilde{C}$, then the membership function of $\tilde{C}$ is defined as $\mu_{\tilde{C}}(z) = \sup_{z=x+y} \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)\}$. It is easy to show that $\tilde{c}_r x (r = 1,\ldots,q)$ and $\tilde{a}_i x (i = 1,\ldots,m)$ are fuzzy numbers too [3]. $\lesssim$ represents a flexible constraint, i.e., smaller violations of constraints could be acceptable, so we admit a tolerance interval for the right hand sides.

In order to work with fuzzy numbers we propose to use a canonical representation of a fuzzy number $\tilde{a}$ through two real indices [2]: the *value* '$V(\tilde{a})$' and the *ambiguity* '$A(\tilde{a})$', that allow for the capturing of relevant information contained in a fuzzy number. The *value* of a fuzzy number may be seen as its central value and the *ambiguity* may be interpreted as the global spread of the number. The *value* and the *ambiguity* are calculated as follows,

$$V(\tilde{a}) = \int_0^1 \alpha[f_a^{-1}(\alpha) + g_a^{-1}(\alpha)]d\alpha, \quad A(\tilde{a}) = \int_0^1 \alpha[g_a^{-1}(\alpha) - f_a^{-1}(\alpha)]d\alpha$$

where f_a and g_a are respectively the left and the right shape function of fuzzy number $\tilde{a}$.

Delgado et al. [2], proposed to rank two fuzzy numbers by the *value-ambiguity* indices. The obtained ranking should be mainly determined by the *value*. Only when the values are almost equal, could the ranking result be driven by the *ambiguity*. According to these considerations the aforementioned authors suggest a lexicographic ranking procedure with the following steps: In the first step the *values* of the fuzzy numbers are compared. If they are approximately equal then one can proceed to

the second step. If this is not the case, the fuzzy numbers are ranked according to the relative position of the *values*. In the second step the *ambiguities* of the 'almost equal' fuzzy numbers are compared. Observe that more ambiguity gives the chance of very good results but, at the same time, the risk of very bad results. That is to say a risk-adverse DM should prefer the fuzzy number with less ambiguity and a risk-seeking DM should prefer the fuzzy number with larger ambiguity. In this paper, the above ranking procedure is introduced in our model in order to solve a flexible programming problem with fuzzy coefficients.

It is easy to show that

$$V(\tilde{a}_i x) = \sum_{j=1}^{n} V(\tilde{a}_{ij}) x_j, \quad A(\tilde{a}_i x) = \sum_{j=1}^{n} A(\tilde{a}_{ij}) x_j$$

and similarly for $\tilde{c}_r x$.

We notice that to figure out the *value-ambiguity* indices for a non triangular-trapezoidal fuzzy number is very easy. So our approach easily permits for handling with fuzzy numbers to have non linear membership functions.

2 Solving a Multiobjective Programming Problem with Fuzzy Coefficients

Following Zimmermann [10], we will assume that the DM can establish in model (1) a fuzzy aspiration level $\tilde{g}_r$ for each objective function. Then it can be written as follows

$$\begin{aligned} &\text{Find } x \\ &\text{such that } \tilde{c}_r x \gtrsim \tilde{g}_r \;\; r = 1, \ldots, q \\ &\qquad\qquad \tilde{a}_i x \lesssim b_i \;\; i = 1, \ldots, m \\ &\qquad\qquad x \geq 0. \end{aligned} \tag{2}$$

The membership function $\mu_k(x)$ represents the degree to which the decision vector x satisfies the k-th fuzzy goal or constraint. Therefore $\mu_k(x)$ should be 0 if the k-th constraint is strongly unfulfilled, and 1 if it is completely satisfied. Then, for flexible inequalities of type $\gtrsim$, $\mu_k(x)$ should increase monotonously from 0 to 1 over the tolerance interval and, for inequalities of type $\lesssim$, $\mu_k(x)$ should decrease monotonously from 1 to 0 over the tolerance interval.

As was stated in the introduction, in order to solve problem (2), we use the *value-ambiguity* ranking procedure following Delgado et al. [2], proposing ourselves a procedure consisting of two steps. In the first step we work with the *value* of the fuzzy coefficients and in the second step we choose, between the almost equal solutions of the first step, those with lesser (or larger) ambiguity (depending on the DM attitude in the face of risk).

(1) First step: working with the value of the fuzzy coefficients

We propose to replace each fuzzy constraint of (2) by one pessimistic crisp one (3) and a new objective (4),

$$\tilde{a}_i x \lesssim b_i \Leftrightarrow \begin{cases} a_i^4 x \leq b_i + t_i \\ \mu_i(x) = \mu_{\tilde{g}_r}(V(\tilde{c}_r x)) \rightarrow \max \end{cases} \tag{4}$$

where $V(\tilde{a}_i x)$ is the *value* of the fuzzy number $\tilde{a}_i x$ and t_i is the tolerance margin of the i-th fuzzy constraint. The crisp constraint (4) assures us that even in the most pessimistic situation the tolerance margin will be not surpassed. Symmetrically for the goals of (2)

$$\tilde{c}_r x \gtrsim \tilde{g}_r \Leftrightarrow \begin{cases} c_r^1 x \geq g_r - d_r \\ \mu_r(x) = \mu_{\tilde{g}_r}(V(\tilde{c}_r x)) \rightarrow \max . \end{cases}$$

Therefore model (2) may be described by the following multi-objective optimization problem

$$\max_{x \in X_U} (\mu_1(x), \ldots, \mu_{q+m}(x)) \tag{5}$$

where $X_U = \{x \in \mathbb{R}_+^n : c_r^1 x \geq g_r - d_r, r = 1 \ldots, q; a_i^4 x \leq b_i + t_i, i = 1, \ldots, m\}$.

Following Bellman and Giertz [1], in this paper, in order to aggregate the constraints satisfaction degrees, the 'minimum' operator is used, but depending on the context other aggregation operators could be used [5]. So the global satisfaction degree may be described by

$$\lambda(x) = \min((\mu_1(x), \ldots, \mu_{q+m}(x))). \tag{6}$$

According to Negoita and Sularia [7] the model (2) is equivalent to

$$\begin{array}{l} \max \lambda \\ \text{subject to } \lambda \leq \mu_{\tilde{g}_r}(V(c_r x)),\ r = 1, \ldots, q \\ \qquad\qquad \lambda \leq \mu_{b_i}(V(a_i x)),\ i = 1, \ldots, m \\ \qquad\qquad x \in X_U . \end{array} \tag{7}$$

When the membership functions $\mu_\bullet$ are linear, the model (7) is a linear programming problem. If the membership functions are nonlinear, then they could be approximated by piecewise linear functions and the model (7) could be formulated as a linear programming problem [4].

(2) Second step: to minimize (or maximize) the ambiguity between the almost equal solutions

In the second step the *ambiguities* of the almost equal solutions of the first step are compared. More ambiguity gives the chance of good results but, at the same time, bigger risk of bad values. On the contrary less ambiguity decreases the risk

of bad values in return for a decrease in the chance of good results. That is to say a risk-adverse DM should prefer the solution with less ambiguity and a risk-seeking DM should prefer those with more ambiguity.

Once the problem (7) has been solved, the question is to determine when we can consider that two solutions are approximately equal. According to Delgado et al. [2] the relation approximately equal is to be constructed in each particular case according to the problem characteristics and the DMs subjectivity. As we are working in an uncertain environment the DM will be more comfortable if she/he can reflect her/his preferences through natural language, so we establish a semantic correspondence for the different degrees of satisfaction [9]. In accordance with Miller [6] and Yager [8] human beings are capable of distinguishing between, remembering and using seven or nine terms. Table 1 shows a scale with seven linguistic labels related with the satisfaction degree of the solution of model (7). Likewise Table 1 shows the numerical interval that we associate with each linguistic label.

Let x^o be an optimal solution of (7) and λ^o the corresponding optimal objective value. We consider as almost equal to x^o any feasible solution having an objective value belonging to the same linguistic label as λ^o. Thus according to (6) the *almost equal* solutions to x^o are those that verify $\lambda_1^o \leq \min((\mu_i(x),\ldots,\mu_{q+m}(x))) \leq \lambda_2^o$, where λ_1^o, λ_2^0 are the bounds of the interval corresponding to the linguistic label of λ^o. But obviously no feasible solution can have a global satisfaction degree bigger than the optimal objective value λ^0, then it is enough to look for the feasible solutions that verify $\mu_i(x) \geq \lambda_1^o, i \in \{1,2,\ldots,q+m\}$. Now, to rank these *almost equal* solutions we will take into account the DM's attitude towards uncertainty, finding those with minimum (or maximum) *ambiguity*. In order to avoid solutions biased towards fuzzy coefficients with larger *value*, we use the relative *ambiguity*, that is to say we divide the *ambiguity* of each fuzzy coefficient by its corresponding *value*. Thus in the second step we solve the following model

$$\begin{aligned} &\min \sum_{r=1}^{m} w_r \sum_{j=1}^{n} \frac{A(\tilde{c}_{rj})}{V(\tilde{c}_{rj})} x_j + \sum_{i=1}^{q} w_i' \sum_{j=1}^{q} \frac{A(\tilde{a}_{ij})}{V(\tilde{a}_{ij})} x_j \\ &\text{subject to } \mu_k(x) \geq \lambda_1^o \quad k = 1,2,\ldots,q+m \\ &x \in X_U \end{aligned} \tag{8}$$

Table 1 Set of seven linguistic labels

Linguistic label	Significance	Interval
l_1	Null	[0,0]
l_2	Very low	[0,0.2]
l_3	Low	[0.2,0.4]
l_4	Average	[0.4,0.6]
l_5	High	[0.6,0.8]
l_6	Very high	[0.8,1]
l_7	Perfect	[1,1]

where the weights $w_r \in [-1,1]$ $(r = 1,\ldots,m)$, $w'_i \in [-1,1]$ $(i = 1,\ldots,q)$ represent *a* measure of the attitude of the DM against the uncertaity, regarding the *r*-th goal or *i*-th constraint, respectively. Thus $w_r > 0$ $(w'_i > 0)$ represents a risk-adverse DM, $w_r < 0$ $(w'_i < 0)$ a risk-seeking DM and $w_r = 0$ $(w'_i = 0)$ a risk-neutral DM.

3 An Illustrative Example

Company C manufactures three products P1, P2 and P3. The production capacity is 5 tons for each product. And, for technical reasons, at least 1 ton of each product must be produced. It is estimated that product P1 yields a profit of about 23 monetary units, P2 yields about 8 monetary units and P3 yields about 15 monetary units per ton. Also, it is pointed out that products P1, P2 and P3 yield respectively 'about 3', 'about 1', and 'about 2 units' of pollution per ton respectively. The total pollution amount should be 'essentially smaller' than 10.5 units, considering that 13 is the tolerance threshold. The company has 50 workers and "should not want to contract more workers", considering that the tolerance threshold is for four workers more. Each ton of P1 needs about 15 days work, each ton of P2 needs about 5 days work and each ton of P3 needs about 5 days work. The company is trying to figure out how many units of products P1, P2 and P3 should be produced to "improve the present benefit situation substantially", which can be expressed as follows: a benefit greater or equal to 95 m.u. is fully satisfactory, a benefit of less than 70 m.u. being unacceptable.

The problem can be expressed by the following model (see (2)):

$$\begin{aligned} &\text{Find } (x_1,x_2,x_3) \\ &\text{Such that } \tilde{23}x_1 \oplus \tilde{8}x_2 \oplus \tilde{15}x_3 \gtrsim 95 \quad \text{(total profit)} \\ &\tilde{3}x_1 \oplus \tilde{1}x_2 \oplus \tilde{2}x_3 \lesssim 10.5 \quad \text{(environmental impact)} \\ &\tilde{15}x_1 \oplus \tilde{5}x_2 \oplus \tilde{5}x_3 \lesssim 50 \quad \text{(employment)} \\ &x_1,x_2,x_3 \leq 5 \quad \text{(production capacity)} \\ &x_1,x_2,x_3 \geq 1 \quad \text{(technical necessities)} \end{aligned} \tag{9}$$

The uncertain parameters are estimated by the following fuzzy numbers, triangular or trapezoidal for the sake of simplicity,

$$\begin{gathered} \tilde{23} = (22.5,22.8,23.4,22.98); \tilde{8} = (7.9,8,8,8.2); \tilde{15} = (14.7,14.9,15.1,15.4); \\ \tilde{3} = (2.9,3,3,3.1); \tilde{1} = (0.95,1,1,1.1); \tilde{2} = (1.9,2,2,2.5); \\ \tilde{15} = (14.5,15,15,15.3); \tilde{5} = (4.8,5,5,5.1); \tilde{5} = (4.9,5,5,5.1) \end{gathered}$$

(a) First step: According to the above considerations, at first priority level we work with the value of fuzzy numbers. Thus we write (see model (7)),

$$
\begin{aligned}
&\max \lambda \\
\text{s.t. } &\lambda \leq (1/20)(22.98x_1 + 8.02x_2 + 15.02x_3 - 70) \\
&\lambda \leq (1/2.5)(13 - 3x_1 - 1.01x_2 - 1.99x_3) \\
&\lambda \leq (1/4)(56 - 14.97x_1 - 4.98x_2 - 5x_3) \\
22.50x_1 + 7.9x_2 + 14.7x_3 &\geq 75 \\
3.1x_1 + 1.1x_2 + 2.05x_3 &\leq 13 \\
15.3x_1 + 5.1x_2 + 5.1x_3 &\leq 54 \\
x_1, x_2, x_3 &\leq 5 \\
x_1, x_2, x_3 &\geq 1.
\end{aligned} \tag{10}
$$

The optimal solution for this model is $x_1 = 1.41, x_2 = 5, x_3 = 1$, whose global satisfaction degree is $\lambda^o = 0.7$.

(b) Second step: According to Table 1 the value $\lambda^o = 0.7$ belongs to the linguistic label l_5, that is to say its global satisfaction degree is 'high', like all solutions that produce a global satisfaction degree between $\lambda_1^o = 0.6$ and $\lambda_2^o = 0.8$. So we consider all of them to be almost equal solutions. Thus according to (8) we solve the following model

$$
\begin{aligned}
\min\, & w_1\left(\frac{0.28}{22.98}x_1 + \frac{0.05}{8.02}x_2 + \frac{0.18}{15.02}x_3\right) + \\
& w_2\left(\frac{0.03}{3}x_1 + \frac{0.03}{1.01}x_2 + \frac{0.03}{1.99}x_3\right) + \\
& w_3\left(\frac{0.13}{14.97}x_1 + \frac{0.05}{4.98}x_2 + \frac{0.03}{5}x_3\right) \\
\text{s.t. } & (1/20)(22.98x_1 + 8.02x_2 + 15.02x_3 - 75) \geq 0.6 \\
& (1/2.5)(13 - 3x_1 - 1.01x_2 - 1.99x_3) \geq 0.6 \\
& (1/4)(54 - 14.97x_1 - 4.98x_2 - 5x_3) \geq 0.6 \\
& 21x_1 + 7.5x_2 + 13x_3 \geq 75 \\
& 3.2x_1 + 1.2x_2 + 2.2x_3 \leq 13 \\
& 15x_1 + 5x_2 + 2x_3 \leq 54 \\
& x_1, x_2, x_3 \leq 5 \\
& x_1, x_2, x_3 \geq 1.
\end{aligned} \tag{11}
$$

Suppose the DM is risk-adverse in the three constraints, then $w_1 = w_2 = w_3$. The optimal solution to the corresponding model is: $x_1 = 2.7, x_2 = 1, x_3 = 1$. This solution, supplies less profit than the solution x^o obtained in the first step (model (11)), in exchange for producing less environmental impact and spending less working days (See Fig. 1).

Now suppose that the DM is risk-seeking in profit and pollution and risk-adverse to working days. Thus, in (11) $w_1 = -1, w_2 = -1, w_3 = 1$. The optimal solution to the corresponding model is: $x_1 = 1, x_2 = 5, x_3 = 1.74$. (See Fig. 2).

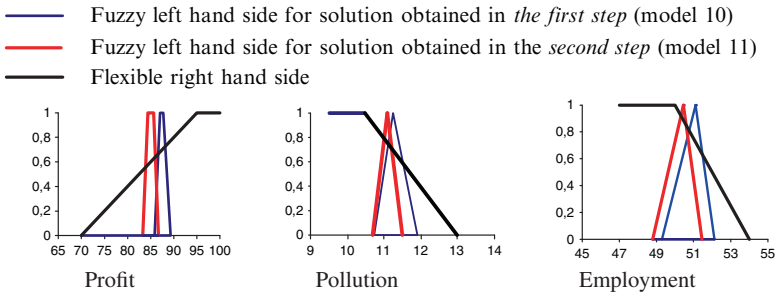


Fig. 1 Solution in case the DM is risk-adverse in the three constraints

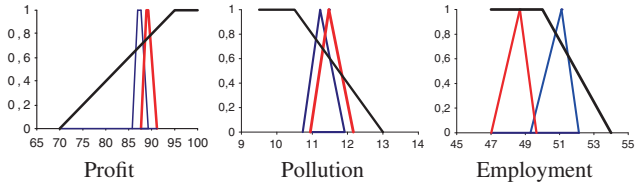


Fig. 2 Solution in case the DM is risk-seeking in profit and pollution and risk-adverse in employment

These graphics allow the DM to visualize the risk of non fulfilment of the constraints and to compare the different almost equal solutions. In the case that she/he doesnt agree with the solutions obtained, she/he can change the tolerance threshold of the flexible constraints or the linguistic labels to her/his degree of satisfaction.

4 Conclusions

Delgado et al. [2] have suggested a lexicographic ranking procedure for the *value-ambiguity* pair to rank fuzzy numbers. In this paper we propose a method that applies this ranking procedure to solving flexible multiobjective linear programming problems with fuzzy coefficients. The use of *value-ambiguity* indices allow us to use nonlinear fuzzy numbers without losing information or increasing the complexity. Our method allows the DM to interact in all steps of the decision-making process. In the first step she/he can determine the tolerance interval of the flexible constraints and the linguistic labels for her/his degree of satisfaction. In the second step she/he can change the weights of the *ambiguities* to generate different almost equal solutions. Lastly our method supplies graphics that allow the DM to visualize the risk of non fulfilments of the constraints and to compare the different almost equal solutions.

References

1. Bellman R, Giertz M (1973) On the analytic formalism of the theory of fuzzy sets. Inform Sci 5:149–156
2. Delgado M, Vila M, Voxman W (1998) On a canonical representation of fuzzy numbers. Fuzzy Sets Syst 93:125–135
3. Dubois D, Prade H (1978) Operations on fuzzy numbers. Int J Syst Sci 9(6):613–626
4. Jiménez M, Arenas M, Bilbao A, Rodríguez Uría M (2005) Approximate resolution of an imprecise goal programming model with nonlinear membership function. Fuzzy Sets Syst 150:129–145
5. Kreinovich V, Nguyen H (2006) Which fuzzy logic is the best: Pragmatic approach (and its theoretical analysis). Fuzzy Sets Syst 157:611–614
6. Miller G (1969) The magical number seven or minus two: some limits on our capacity of processing information. Psychol Rev 63:81–97
7. Negoita C, Sularia M (1976) On fuzzy mathematical programming and tolerances in planning. Econ Comput Econ Cybern Stud Res 3:83–95
8. Yager R (1993) Families of OWA operators. Fuzzy Sets Syst 59:125–148
9. Zadeh L (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst 1:3–28
10. Zimmermann H (1978) Fuzzy programming and linear programming with several objective functions. Fuzzy Sets Syst 1(1):45–55

A Robust-Solution-Based Methodology to Solve Multiple-Objective Problems with Uncertainty

Daniel Salazar, Xavier Gandibleux, Julien Jorge, and Marc Sevaux

Abstract This paper presents the formulation and evaluation of a methodology to solve multiple-objective problems in the presence of uncertainty based on robustness. The methods were developed regarding a particular real "Released when Completing blocking (RCb)" scheduling flowshop problem characterized by several equivalent solutions, but they can be extended to another robustness problems.

Keywords: Flowshop scheduling problem · Genetic algorithm · Multiobjective evolutionary algorithm · Multiobjective simulated annealing · Robustness · Robust schedule · Release when completing blocking

1 Introduction

Robustness is a very important concept present in diverse areas related to Multicriteria Decision-Making. In general words we can say that robustness is an attribute related to the "ability of a subject to cope well with uncertainties" [12], more precisely the uncertainties that accompany the input of a system. In this work we are particularly interested in the concept of robust solution in optimisation and its application to find the final set of alternatives in Multiple-Objective (MO) problems. The application of robustness described here is framed in a specific "Released when Completing blocking"(RCb) bi-objective hybrid flowshop scheduling problem, but the methodologies studied can be adapted to other classes of problems.

D. Salazar (✉)
IUSIANI, University of Las Palmas de Gran Canaria Edif. Polivalente. Campus de Tafira, CP 35017 Las Palmas, Spain
e-mail: danielsalazaraponte@gmail.com

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

2 The RCb Hybrid Flowshop Bi-Objective Scheduling Problem

This problem was studied first in [3] and is originated in a waste treatment plant. It consists of a set of silos (1^{st} stage; $m \geq 1$ identical parallel machines) where a group of trucks unload the waste (1^{st} operation), and a mixer (2^{nd} stage; critical machine) that processes the waste to dispose it of. There is no storage capacity, thus once a silo accepts the load it cannot be released until the waste is completely transferred to the mixer and the processing ends. Subsequently, the sequence strictly begins at the 1^{st} stage and finalizes at the 2^{nd} with no preemption (see Fig. 1).

This problem poses two objectives, the minimisation of the Makespan -strongly associated with the critical machine- and the minimisation of the Average Waiting Time between the ready time (when a truck arrives) and the starting time (when the unloading begins) of each job at the 1^{st} stage. No other objectives or constraint were supplied.

The optimisation of these two objectives was tackled in [3] with a Multiple-Objective Simulated Annealing algorithm (see Algorithm 1) solving two real instances (see Fig. 2) -corresponding to a waste treatment problem- along with several random instances . The procedure was built using MNEH, an heuristic proposed by Nawaz, Enscore and Ham and modified by Martínez [7] (see [3] for more details).

Contrary to what was expected, the output is featured by few or even only one non-dominated point and a large number of equivalent schedules, no matter what instance is chosen. In other words, there are several different schedules that show the same Makespan and the same Average Waiting Time: different alternatives for the DM that yield the same performance in terms of objectives values.

The results obtained by [3] show that the algorithm is capable of solving the optimisation problem, so one could say that the problem itself does not represent a technical challenge. Nevertheless the described situation raises an unfavourable panorama for decision-making: once a desired solution is identified in the objective

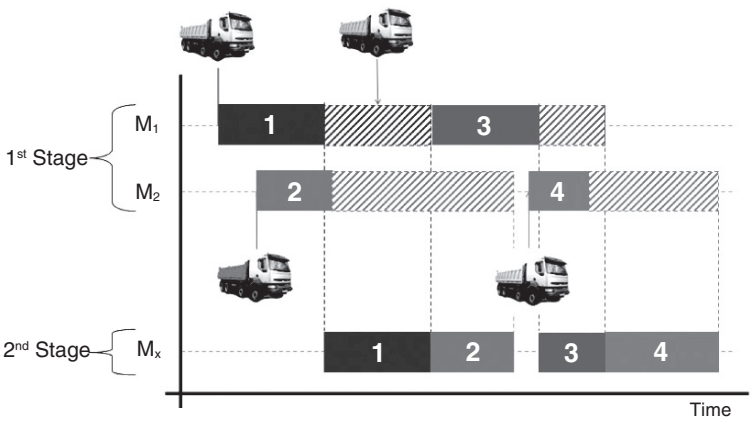


Fig. 1 RCb scheduling problem in a waste treatment plant (two stages example)

Algorithm 1 Multiple-Objective Simulated Annealing [3]

begin
  $X_{PE} \leftarrow \emptyset$; $x_{init} \leftarrow$ `xMNEH()`
  for all $\lambda \in \Lambda$ **loop**
    $T_0 \leftarrow T$; $N_{count} \leftarrow 0$
    $n \leftarrow 0$; $x_n \leftarrow x_{init}$; $X_{PE_\lambda} \leftarrow \{x_n\}$
    repeat randomly draw $x \in \mathcal{N}(x_n)$
      if `isBetter`(x, x_n)
        or else `isAccepted`$(x, x_n, n, T_n, \lambda)$
      then
        $X_{PE_\lambda} \leftarrow$ `archive`(X_{PE_λ}, x);
        $x_{n+1} \leftarrow x$; $N_{count} \leftarrow 0$
      else
        $x_{n+1} \leftarrow x_n$; $N_{count}++$
      endIf
      $n++$
      `updateParameters`$(\alpha, n, T_n, N_{count})$
    until `isFinished`(T_n)
    $x_{init} \leftarrow$ `xMinAttente`(X_{PE_λ})
  endLoop
  $X_{PE} \leftarrow$ `merge`(X_{PE_λ})
end

where:

- `isBetter`, is a procedure that calculates $\Delta z_k \leftarrow z_k(x) - z_k(x_n)$ and returns $\neg(\forall k : \Delta z_k \geq 0)$.
- `isAccepted`, is a procedure that calculates the probability of acceptance $prob_n$ for the current iteration with $\exp\left(-\sum_{k=1}^{q} \lambda_k \ (z_k(x) - z_k(x_n))/T_n\right)$, and then generates a random number $prob$ uniformly distribute in $[0,1]$ and returns $(prob < prob_n)$.
- `isFinished`, is a procedure that verifies if whether the predefined number of interation N_{stop} or the limit temperature for the cooling process T_{stop} has been reached.
- `merge`, is a procedure that merges $X_{PE_\lambda}, \lambda \in \Lambda$ into X_{PE} and eliminates the dominated solutions

space, how does one select the right alternative from the numerous set of equivalent schedules? In other words, can a DM choose a particular schedule from hundreds of equivalent solutions? A random selection is does possible but isn't it risky? It is possible to refine the final set to facilitate the decision making?

To help answer these questions we incorporate the concept of robustness presented in the next sections, based on the following considerations:

- **It is a real life problem:**
  - Its implementation entails dealing with uncertainty.
  - The amount of information about the problem is scarce.
- **We propose:**
  - To offer the DM optimal and robust solutions.
- **We assumed:**
  - Trucks may arrive earlier or later the expected time.

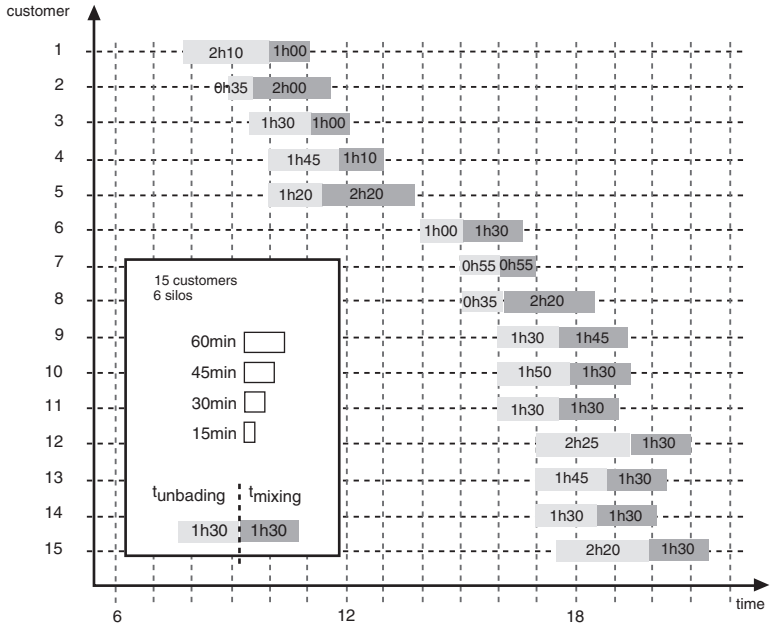


Fig. 2 An example of a real instance: ready and processing time of each truck

– No online scheduling is possible.
– Delays in tasks are not considered.
– The sequence will be strictly respected.
– The DM is averse to giving more information.

3 Robustness in Optimisation

There are several formulations of robustness closely related to design. Most of them turn around the idea of insensitivity of a solution or design to small variation of the decision variables and/or the design parameters [5, 13, 14]. Nevertheless, none of them establishes how insensitive a solution should be or what "small variation"means. Since the answers to these questions rely on the Decision Maker (DM), it is possible to classify the robustness concepts in terms of the type of information provided by an analyst/DM and the stage where it is provided.

Let us consider any optimisation procedure as a three stages process: (1) input, (2) solving, (3) output. The first step comprises the acquisition of data, the modelling of the objectives and disturbances and the exercise of *a priori* DM's preferences. The second step consists in optimising the function(s) by means of a classical and/or a heuristic method; and finally in the third step the DM obtains the efficient frontier from which to choose the desired alternative. Here we focus our analysis to

metaheuristics. According to this, it is possible to identify in the literature two basic approaches regarding either single or MO.

Let X be a vector of decision variables and let δ be a vector of disturbances. Let F(X) be a function that describes the performance of a system, and finally let F_t be a target value.

1^{st} **Approach: from input to output** This is the typical approach adopted in robust design which is strongly associated with Taguchi's work [2]. Two subproblems derive from this approach:

(a) Minimize the deviation:

$$\min_{x} \operatorname*{var}_{x,\sigma} |F_t - F(X+\delta)|$$

(b) Optimize F(X) and minimize the variability:

$$\operatorname*{Opt}_{x} F(X) \wedge \min_{X} \operatorname*{var}_{X,\sigma} |F(X) - F(X+\delta)|$$

2^{nd} **Approach: from output to input** This is the approach adopted in [8, 9, 11] and consists in assessing that solution that allows or resists to the largest level uncertainty in the input without missing any requirement.

1^{st} Approach:

In evolutionary optimisation, this approach is commonly accomplished by means of an Effective Function (either in single [4, 14] and multiple objective optimisation [1]) formulated as $F_{eff} = \frac{1}{N}\sum_{i=1}^{N} F(X+\delta_i)$ where F is the objective function which is averaged over a neighbourhood of the decision vector X. Here the analyst/DM points out a plausible assumption or real information (Probability Distribution Functions or PDF) to generate N realizations (δ_i) of the disturbance δ before searching the optimum, but none requisite is stated for the output $F_{eff}(X)$. Notice that the main drawback in terms of robustness of the solution is that the problem cannot be solved without an appropriate description of the uncertainty, i.e. a way of generating the disturbances δ.

2^{nd} Approach:

In evolutionary optimisation, this approach adopts a different point of view, using metaheuristics to search for the maximal range of each disturbance (refers as *Maximal volume Inner Box* -MIB- in [8]) that assures the objective functions always meet the prerequisites set by the DM [8]. This approach has been extended by formulating the original problem as a multiple objective problem [9, 11], by converting one constraint into an objective. However, problems subject to uncertainty more than one objective had not been tackled before.

One way to follow this approach in optimisation consists simply in settings some goals or requirements to the objective function, in that way that the performance is bounded, yielding $F^{\ell} \leq F(X+\delta) \leq F^{\upsilon}$. The bounds are fixed by some DM's criteria and therefore the problem converts into an optimisation problem of maximising the size of the deviation s.t. the original constraints plus the performance requirement $F^{\ell} \leq F(X+\delta) \leq F^{\upsilon}$. Notice that this approach cannot be applied without some previous knowledge of the range of F(X).

4 Proposed Methodology

As the DM did not provide any information about δ nor about plausible limits F^{ℓ}, F^{υ}, it was not possible to solve the problem with a single application of any of the robustness approaches presented earlier. Given that panorama, we propose a mixed strategy that merges the two approaches, based on the least number of assumptions that derive from answering these questions: What is the maximal dimension that can be neglected in the objective space? And, what is a reasonably significant level of uncertainty in the input?

4.1 Formulation

The answer to the first aforementioned question gives the *maximal level of indifference* that the DM is willing to tolerate, i.e. any difference lower than this level is neglected. The second answer gives the minimal level of uncertainty that is considered significant in the input. Thus, with the two quantities we developed the following methods (Fig. 3):

Method 1:

Step 1: Define a set of measures ε_i each of one representing the maximal level of indifference tolerated by the DM for each objective (we use here the concept of ε-dominance but based on preference information)

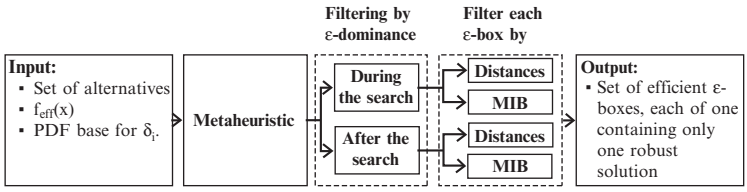


Fig. 3 Alternatives for Robust-Solution-based MO Optimisation

Step 2: Optimize with robust values as $F_{eff}(X) = (F^1_{eff}(X), F^2_{eff}(X))$

- Let any schedule S be defined by S(X) = $\{\triangle, T_S(X)\}$ a sequence $\triangle$ and a vector of starting times T_S
- Where the Effective Makespan is

$$F^1_{eff}(X) = \frac{1}{N}\sum^{N} makespan\,(S(X+\delta))$$

and the Effective Average Waiting Time is

$$F^2_{eff}(X) = \frac{1}{N}\sum^{N} AWT\,(S(X+\delta))$$

and $\triangle$ remains the same and $S(X+\delta) = \{\triangle, T_S(X+\delta)\}$ for $\delta_i = U(\pm 10)$

Step 3: Wipe out all the solutions located in dominated ε-hyperboxes, and prune the set of non-dominated solutions, selecting only one solution for each non-dominated hyperbox according to some criterion.

Possible criteria:

- Distance to the ε-hyperbox ideal vertex, or
- MIB = Maximal Volume Inner Box

The core procedure is based on the use of effective functions. For that matter a base PDF is assumed which represents the minimal significant level of uncertainty that any schedule should be able to cope with. Additionally, the concept of ε-dominance [6] is incorporated as a filtering tool as in [10], in such a way that the objective space is divided into ε-boxes and all solutions contained into each ε-box are indifferent to the DM.

In order to filter the content of the non-dominated ε-boxes, we tested two criteria: the former consists in removing all the solutions but the closer to the ideal point of each box, whereas the second keeps the solution with the larger MIB. The filtering process is then performed during the whole search.

Method 2:

Methods 1 and 2 employ the same approaches of robustness, the main differences relies on the stage when the 2^{nd} Approach (MIB) is applied. In Method 1 the pruning phase inside each hyperbox is performed during the search either for the distance criterion or the MIB criterion. In Method 2 there is no pruning phase, thus the whole set of non-dominated solutions found are presented to the DM, then they can select the hyperbox of interest and employ the MIB criteria to decide which solution to select.

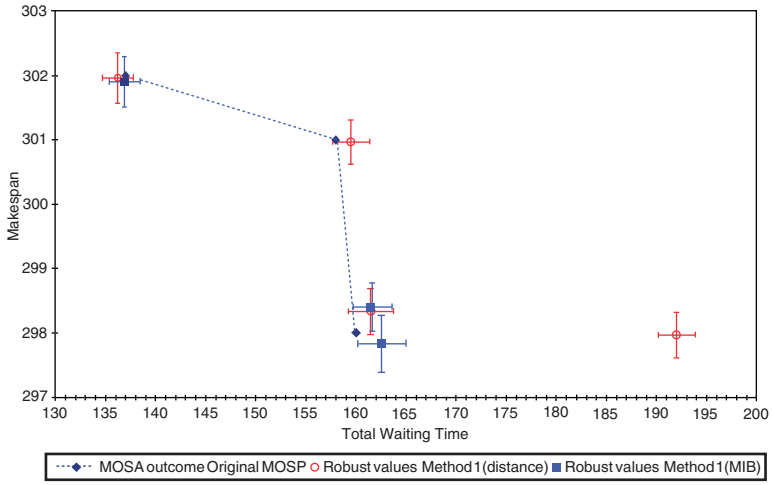


Fig. 4 Results obtained with Method 1 using the two criteria for pruning

4.2 Implementation Details

The implementation for solving our RCb MO problem is based on the assumption that no online scheduling can be done, therefore the sequence of jobs is invariable and only the ready and starting times can change. As PDF base we chose an uniform distribution $U_j(read\,time_j \pm \delta_j)$, where δ_j=10' for each job j. With these assumptions, once a schedule is generated by MOSA (Algorithm 1), we can assess the effective objective functions (Makespan and Average Waiting Time) (F^1_{eff}, F^2_{eff}) by means of a simulation of 30 samples, yielding good results. The simulation follows the original sequence, readjusting only the starting times. Afterward the determination of the ε-box the vector (F^1_{eff}, F^2_{eff}) belongs to is straightforward. We use additive ε-dominance [5] and ε_i=10'.

The calculation of the MIB of any particular schedule is performed by a genetic algorithm (GA) of 20 individuals in 30 generations, where each individual represents a vector of j maximal disturbances δ_j initialized as δ_j=10' and has a potential MIB associated, assessed as $\prod_j(\delta_j)$. The GA looks for the largest ranges of δ_j (maximizes MIB) that assure (F^1_{eff}, F^2_{eff}) still belongs to its original ε-box or to an immediate neighbour. For each individual (F^1_{eff}, F^2_{eff}) is recalculated.

5 Results

The different alternatives to conjugate the robustness approaches described earlier where tested and evaluated with the real instances of data. Fig. 4 shows the results of applying Method 1 with pruning by distances(circles) and pruning by MIB (solid

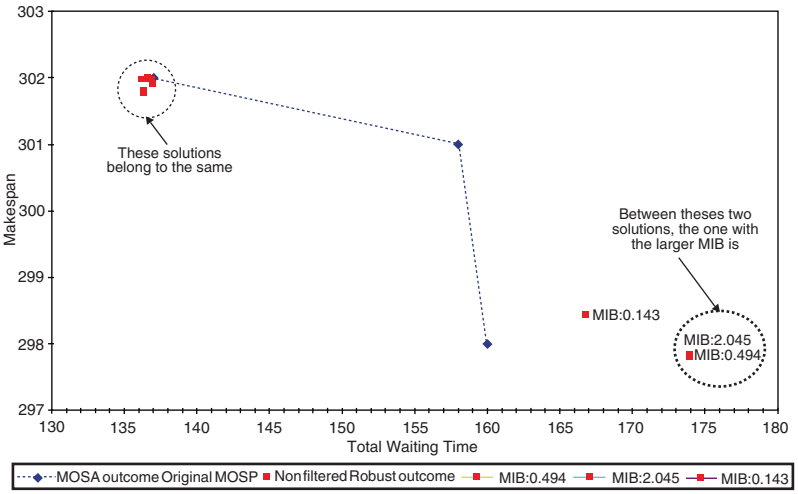


Fig. 5 Results obtained with Method 2

squares). Notice that the two approaches produce similar outcomes and show the ability to reduce the original huge number of equivalent solutions to a few robust non-dominated schedules by means of a logical procedure. Nevertheless, in this case pruning by distances seems a better option, since it produces a better spread outcome with a lower computational burden.

However, a pretty different appreciation can be obtained when we apply Method 2 (Fig. 5). The upper left cluster represents a groups of solutions belonging the same ε-box. It is evident that in this case a pruning by distances will favor the solution placed at the lower left "corner"of the cluster. That solution seems to overcome even the non-dominated point obtained without considering robustness (solid rhomboid). Nevertheless this apparent gain on quality is an effect of the Monte Carlo simulation. In other words, we don't have crispy values but intervals, as it was represented in Fig. 4. Thus, comparing distances could be even not relevant for that case. Nonetheless, two close non-dominated points might have very different MIB as is shown in the lower right cluster of Fig. 5. As a matter of fact, the solutions practically overlap each other but one of them is five times more robust that the other. It shows that a single application of an effective function may underestimate the robustness of a particular solution regarding the others, especially due to the imprecision introduced by the simulation or average procedure.

It is important to mention that the methods developed and tested in this work may be adapted to diverse real problems where the information is scarce. The results obtained lead us to conclude that the MIB should be included to improve assure a higher quality of the results. Nevertheless, since its application entails a considerable computational burden, some balance must be found between Method 1 and Method 2 using pruning by MIB, in order to improve the efficiency. This step is open for further research.

6 Conclusions

The proposed methodology permits to reduce the huge number of equivalent solutions related to each non-dominated point by means of a logical procedure based on two different formulations of the concept of robustness, as well as the ε-dominance merged with preferences. The results shown that the MIB measure is a remarkable complement of the F_{eff} approach since it allows the DM to find even more robust solutions that the single application of F_{eff}, without sacrifice the attained level of optimality. In that sense, filtering by MIB seems a better alternative than filtering by distance.

Finally, the present work poses new issues for further researches like the generalization of the current approach to other kind of problems, the evaluation of the effect of executing the filtering process during or after the search as well as the convenience of incorporating MIB as an additional objective. Some of these questions are currently under study.

References

1. Deb K, Gupta H (2005) Searching for robust Pareto-optimal solutions in multi-objective optimization. In: Evolutionary Multi-Criterion Optimization, LNCS 3410. Springer, Berlin, Germany, pp 150–164
2. Dehnad K (ed) (1989) Quality control, robust design, and the taguchi method. Wadsworth and Books, Pacific Grove
3. Gandibleux X, Jorge J, Martínez S, Sauer N (2006) Premier Retour d'Experience sur le Flow-Shop Biobjectif et Hybride a Deux Etages avec une Contrainte de Blocage Particuliere. In: 6^e Conférence Francophone de MOdélisation et SIMulation (MOSIM'06), Rabat, Morocco, April 2006
4. Jin Y, Branke J (2005) Evolutionary optimization in uncertainty environments - a survey. IEEE Trans Evol Comput 9(3):303–317
5. Jin Y, Sendhoff B (2003) Trade-off between optimality and robustness: An evolutionary multiobjective approach. In: Evolutionary Multi-Criterion Optimization, LNCS 2632. Springer, Berlin, Germany, pp 237–251
6. Laumanns M, Thiele L, Zitzler E, Deb K (2002) Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), San Francisco, California, USA, July 2002. Morgan Kaufmann, New York, pp 439–447
7. Martínez S (2005) *Ordonnancement de Systèmes de Production avec contraintes de blocage*. Dissertation, University of Nantes, France
8. Rocco C (2005) A hybrid approach based on evolutionary strategies and interval arithmetic to perform robust designs. In: Applications of Evolutionary Computing, EvoWorkshops2005, LNCS 3449, Lausanne, Switzerland, 30 March–1 April 2005, pp 623–628
9. Rocco C, Salazar D (2007) A hybrid approach based on evolutionary strategies and interval arithmetic to perform robust designs. In: Yang S, Ong Y-S, Jin Y (eds) Evolutionary Computation in Dynamic and Uncertain Environments, vol 51 of Studies in Computational Intelligence (SCI), Chap. 24. Springer, Berlin, pp 543–564
10. Salazar DE, Martorell SS, Galván BJ (2005) Analysis of representation alternatives for a multiple-objective floating bound scheduling problem of a nuclear power plant safety system.

In: Evolutionary and Deterministic Methods for Design,Optimisation and Control with Applications to Industrial and Societal Problems EUROGEN 2005, September 12–14, Munich, Germany
11. Salazar D, Rocco S (2007) Solving advanced multi-objective robust designs by means of multiple objective evolutionary algorithms (MOEA): A reliability application. Reliab Eng Sys Saf 92(6):697–706
12. Sayin S (2005) Robustness analysis: robustness algorithms for multiple criteria optimization. Newslett Eur Working Group "Multiple Criteria Decision Aiding" Ser 3, No 11
13. Sevaux M, Sörensen K (2004) Robustness analysis: optimisation. Newslett Eur Working Group "Multiple Criteria Decision Aiding" Ser 3, No 10
14. Tsuitsui S, Ghosh A (1997) Genetic algorithms with a robust solution searching scheme. IEEE Trans Evol Comput 1(3):201–208

On the Use of Preferential Weights in Interactive Reference Point Based Methods

Kaisa Miettinen, Petri Eskelinen, Mariano Luque, and Francisco Ruiz

Abstract We introduce a new way of utilizing preference information specified by the decision maker in interactive reference point based methods. A reference point consists of aspiration levels for each objective function. We take the desires of the decision maker into account more closely when projecting the reference point to become nondominated. In this way we can support the decision maker in finding the most satisfactory solutions faster. In practice, we adjust the weights in the achievement scalarizing function that projects the reference point. We demonstrate our idea with an example and we summarize results of computational tests that support the efficiency of the idea proposed.

Keywords: Interactive methods · Multiple objectives · Multiobjective optimization · Multiobjective programming · Preferences · Reference point methods

1 Introduction

In multiobjective optimization, several objective functions are to be optimized simultaneously. Because the objective functions typically are conflicting, it is impossible to find a solution where all the objectives can simultaneously reach their individual optima. Instead, we can identify compromise solutions, that is, so-called nondominated solutions, where none of the objectives can get a better value without deterioration to at least one of the other objectives. Ultimately, the task of solving multiobjective optimization problems is to find the best nondominated solutions to be called a final solution. This usually necessitates additional information from a decision maker (DM), an expert in the domain of the problem in question.

K. Miettinen (✉)
Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
e-mail: kaisa.miettinen@jyu.fi

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

Many methods have been developed for solving multiobjective optimization problems during the years. Among them, interactive multiobjective optimization methods are widely used (see, e.g. [6] and references therein). In them, a solution pattern is formed and iteratively repeated, and the DM takes actively part in the solution process by specifying and refining preference information. In this way, the DM can learn about the possibilities and limitations of the problem and about the interdependencies among the objective functions. Furthermore, only such nondominated solutions are generated that are interesting to the DM. Assuming the DM has time enough to take part in an interactive solution process, the final solution can be expected to be satisfactory because the DM can genuinely affect and direct the solution process in order to find a desired final solution.

There are many interactive methods and they differ basically from each other in what kind of information is asked from and shown to the DM at each iteration, and in the way the successive solution candidates are calculated. Examples of types of preference information asked from the DM include marginal rates of substitution, surrogate values for trade-offs, classification of objective functions and reference points. For further details, see, for example, [6, 14] and references therein.

Among interactive approaches, methods using reference points (for the idea see, e.g. [6, 15]) have been popular (for some comparative studies, see, e.g. [9, 10]) because of their straightforward nature. A reference point consists of desirable values for each objective function. For DMs, reference points are a natural way of expressing desires in solutions because DMs do not have to learn to use new, artificial concepts. Instead, objective function values are used that as such are meaningful and understandable for DMs. Examples of methods utilizing reference points include reference point method [15], visual interactive approach [4], STOM [13], GUESS [2] and light beam search [3]. In addition, methods based on classification are closely related to reference point methods because with a classification, the DM indicates what kind of changes are desirable in the current objective function values. Thus, a reference point can be formed once a classification has been made [11]. Methods based on classification can be found, for example, in [1, 9, 11, 13], among others.

In this paper, we concentrate on interactive reference point based methods where, as already mentioned, the DM is at every iteration asked to specify a reference point consisting of desirable or acceptable values for each objective function. The next solution candidate is then generated by minimizing an achievement scalarizing function. In practice this means that the reference point is projected to the set of nondominated solutions and any nondominated solution can be found by altering the reference point. In most of the methods using achievement scalarizing functions, while the reference point is changed at each iteration, weights determining the projection direction are kept unaltered during the whole process and their purpose is mainly to normalize different objectives. In all, in achievement scalarizing functions widely used, the weights have no real preferencial meaning. Rather than that, they are just instrumental.

Widely-used interactive reference point based methods are comfortable and intuitive for DMs, and many real applications show that they perform well and,

eventually, are able to find a good solution. Nevertheless, sometimes it may be difficult for the DM to find certain solutions. For example, if the DM has a greater interest in achieving a certain level for a given objective function than for the others, the only way to do it may be to provide much better values to the corresponding aspiration level. In some cases, it may even be necessary to give an aspiration level better than the ideal value for this objective, in order to push the solution towards the desired value. In these cases, the reference point may not have a clear interpretation for the DM. The use of a greater weight for this particular objective would make the process much easier. We believe that, in general, the use of some kind of weights reflecting preferences can ease and accelerate the solution process.

The reference point can be projected in many directions to become nondominated and some of the directions may be more desirable to the DM than others (especially when aspiration levels are unachievable and the reference point is far from the set of nondominated solutions). Because DMs do not usually want to use too much time in the solution process it is important to help the DM in finding a satisfactory solution fast. Our goal is to reflect the DM's desire to reach aspiration levels by incorporating preference information into weights in the achievement scalarizing function. This should result in a solution that is closer to the most preferred solution of the DM.

2 Concepts and Notations

We consider *multiobjective optimization problems* of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}),\ldots,f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x}\in S \end{array} \tag{1}$$

involving k (≥ 2) conflicting *objective functions* $f_i : S \to \mathbf{R}$ that we want to minimize simultaneously. The *decision variables* $\mathbf{x}$ belong to the nonempty compact *feasible region* $S \subseteq \mathbf{R}^n$. *Objective vectors* consist of *objective values* $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ldots,f_k(\mathbf{x}))^T$ and we denote $Z = \mathbf{f}(S)$.

In multiobjective optimization, objective vectors are optimal if none of their components can be improved without deteriorating at least one of the others. More precisely, a decision vector $\mathbf{x}' \in S$ is said to be *efficient* if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i = 1,\ldots,k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ for at least one index j. On the other hand, a decision vector $\mathbf{x}' \in S$ is said to be *weakly efficient* for problem (1) if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}')$ for all $i = 1,\ldots,k$. The corresponding objective vectors $\mathbf{f}(\mathbf{x})$ are called *(weakly) nondominated objective vectors*. Note that the set of nondominated solutions is a subset of weakly nondominated solutions.

Let us assume that for problem (1) the set of nondominated objective vectors contains more than one vector. We calculate the *ideal objective vector* $\mathbf{z}^\star = (z_1^\star,\ldots,z_k^\star)^T \in \mathbf{R}^k$ by minimizing each objective function individually in S, that is,

$z_i^\star = \min_{\mathbf{x}\in S} f_i(\mathbf{x}) = \min_{\mathbf{x}\in E} f_i(\mathbf{x})$ for all $i = 1,\ldots,k$, where E is the set of efficient solutions. This gives lower bounds for the objectives. The upper bounds, that is, the *nadir objective vector* $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}},\ldots,z_k^{\text{nad}})^T$, can be defined as $z_i^{\text{nad}} = \max_{\mathbf{x}\in E} f_i(\mathbf{x})$ for all $i = 1,\ldots,k$. In practice, the nadir objective vector is usually difficult to obtain. Its components can be approximated using a pay-off table but in general this kind of an estimate is not necessarily too good (see, e.g., [6] and references therein.)

Furthermore, sometimes a *utopian objective vector* $\mathbf{z}^{\star\star}$ is defined as a vector strictly better than the ideal objective vector. Then we set $z_i^{\star\star} = z_i^\star - \varepsilon$ for all $i = 1,\ldots,k$, where $\varepsilon > 0$ is a small real number. This vector can be considered instead of an ideal objective vector in order to avoid the case where ideal and nadir values are equal or very close to each other. In what follows, we assume that the set of nondominated objective vectors is bounded and that we have global estimates of the ideal and nadir objective vectors available.

All nondominated solutions can be regarded as equally desirable in the mathematical sense and we need a *decision maker* (DM) to identify the most preferred one among them. A DM is a person who can express preference information related to the conflicting objectives and we assume that less is preferred to more in each objective for her/him. Here we assume that the DM specifies preferences in the form of reference points.

Typically, when solving multiobjective optimization problems, the multiple objective functions and preferences specified by the DM are combined in real-valued *scalarizing functions*. Scalarizing functions can be optimized with appropriate single objective optimization techniques and they generate (weakly) nondominated solutions for the original problem.

The main scheme of interactive techniques based on reference points is the following. At each iteration h, the DM provides aspiration levels q_i^h for every objective f_i $(i = 1,\ldots,k)$, and these levels constitute a reference point $\mathbf{q}^h = (q_1^h,\ldots,q_k^h)^T$ reflecting her/his hopes. Next, an achievement (scalarizing) function is minimized in order to find a solution that best satisfies the hopes expressed. The DM can then give a new reference point and the iterative solution process continues until the DM has found the most preferred solution as the final solution and wants to stop.

An example of an achievement function is given in problem

$$\begin{aligned} &\text{minimize} && \max_{i=1,\ldots,k} \left[\mu_i^h (f_i(\mathbf{x}) - q_i^h) \right] \\ &\text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{2}$$

where μ_i^h is a weight assigned to the objective function f_i. The solution of problem (2) at iteration h is denoted by $\mathbf{x}^h$ and the corresponding objective vector by $\mathbf{f}^h = \mathbf{f}(\mathbf{x}^h)$. The solution is (weakly) efficient for any reference point (see, e.g. [6]). There are also other forms of achievement functions see, for example, [6, 15].

Usually, in reference point based methods the reference point is changed at each iteration, while the weights are kept unaltered during the whole interactive solution process. The weights can be set for all $i = 1,\ldots,k$, for example, as $\mu_i = 1/[z_i^{\text{nad}} - z_i^{\star\star}]$. These weights normalize the values of each objective function

f_i to an approximately similar magnitude with the other objectives. In what follows, we refer to these scaling factors as *basic weights* μ_b.

No matter which achievement function formulation is used, the idea is the same: if $\mathbf{q}^h \in Z + \mathbf{R}^k_+$, then the minimization of the achievement function subject to the feasible region allocates slack between the reference point and nondominated solutions producing a nondominated solution. Here $\mathbf{R}^k_+ = \{\mathbf{q} \in \mathbf{R}^k \mid q_i \geq 0 \text{ for } i = 1,\ldots,k\}$. In other words, in this case the reference point is a nondominated solution of the problem in question or it is dominated by some nondominated solution. On the other hand, if $\mathbf{q}^h \notin Z + \mathbf{R}^k_+$, then the minimization must produce a solution that minimizes the distance between $\mathbf{q}^h + \mathbf{R}^k_+$ and Z, see [6, 15]. In what follows, we say that a *reference point is feasible* if $\mathbf{q}^h \in Z + \mathbf{R}^k_+$. Otherwise, we say that it is *infeasible*. We can easily judge the feasibility of the reference point by studying the sign of the optimal achievement function value.

Let us point out that even though we in the following sections refer to formulation (2), the scheme presented does not depend on the form of the achievement function used and any other formulation could be used as well.

3 Reflecting Preference Information

When in reference point based methods the DM provides at iteration h a new reference point $\mathbf{q}^h$, (s)he may expect that there exists a nondominated objective vector, or a solution very close to it. However, the expectations of the DM may be too optimistic (or pessimistic) and the reference point given may actually be quite far from the set of nondominated objective vectors.

In this section, we propose a scheme to incorporate the DM's preference information to weights in reference point based interactive procedures. We suggest the new scheme to be used so that both the solutions calculated by minimizing the achievement function with the basic weights and with the new weights proposed are shown to the DM. This is because we do not claim that the new weights could in all possible situations give a more preferred solution than the one produced with basic weights.

We assume that the DM is able to rank the relative importance of achieving each aspiration level every time a new aspiration level has been provided. It should be noted here that the DM is not asked to give any global preference ranking of the objectives, but we are interested in the local importance of achieving each of the aspiration levels.

After the DM has specified her/his reference point, (s)he assigns objective functions to classes in an increasing order of importance for achieving corresponding aspiration level. This importance evaluation allows us to allocate the k objective functions into index sets J_r which represent the importance levels $r = 1,\ldots,s$, where $1 \leq s \leq k$. If $r < t$, then achieving the aspiration levels of objective functions in the index set J_r is less important than achieving aspiration levels of the objectives in J_t. One objective function can only belong to one index set but several objectives

can be assigned to the same index set J_r. This means that achieving their aspiration levels is equally important. Here we have different weights depending on whether the reference point is feasible or not.

If the current reference point is infeasible, the weights for objectives f_i with $i \in J_r$ are set as

$$\mu_i^h = r/[z_i^{\text{nad}} - z_i^{\star\star}] \tag{3}$$

for each $r = 1, \ldots, s$. On the other hand, if the reference point is feasible, we set $\mu_i^h = 1/[r(z_i^{\text{nad}} - z_i^{\star\star})]$ for $i \in J_r$ and $r = 1, \ldots, s$. This scheme allows us to set weights using the information given by the DM in order to produce a solution that is closer to the preferences. Note that the number of importance levels s may be different from one iteration to another. For further details, see [5]. As mentioned in the beginning of this section, we propose to calculate the solution together with the one obtained using basic weights. In this way, the DM is able to choose the most preferred solution. Finally, let us remind that the preference schemes suggested can be combined with any achievement function, because we only modify the weights used.

4 Example

We illustrate the behaviour of our weighting scheme with a nonlinear multiobjective optimization problem involving the first two objective functions of the problem described in [7]. For this problem, we have the ideal and nadir objective vectors as $\mathbf{z}^\star = (-8.13, -8.13)^T$ and $\mathbf{z}^{\text{nad}} = (-0.05, 2.02)^T$, respectively. The basic weights are then $\mu_b = (0.12, 0.10)^T$.

Let us now demonstrate how our weighting scheme behaves. We assume that the DM has provided an infeasible reference point $\mathbf{q}^1 = (-6.00, -6.00)^T$ and the preference order ranking 2, 1 for achieving the aspiration levels, that is, it is more important to achieve the aspiration level of the first objective function. Then the weights defined by (3) are $\mu^1 = (0.24, 0.10)^T$. When problem (2) is solved with this information, we obtain $\mathbf{f}^1 = (-4.72, -2.79)^T$. On the other hand, if we had used the basic weights, we would have obtained solution $\mathbf{f}_b^1 = (-3.93, -3.40)^T$, which has a higher, that is, worse value for f_1.

We can also demonstrate what happens if the DM specifies a feasible reference point. Let us assume that the DM sets $\mathbf{q}^2 = (-5.00, 1.00)^T$. In this case, the weights are $\mu^2 = (0.06, 0.10)^T$ and the solution obtained is $\mathbf{f}^2 = (-7.01, -0.26)^T$. If we had used basic weights, we would have obtained solution $\mathbf{f}_b^2 = (-6.53, -0.92)^T$ where, again, the value of f_1 is worse. The solutions are depicted in Fig. 1.

In Figs. 2 and 3 we demonstrate complete interactive solution process with both our new weighting scheme and basic weights, respectively. As can be seen, less (5 vs. 8) reference points, that is, iterations were needed in finding the most preferred solution with the new weighting scheme.

Fig. 1 Example problem

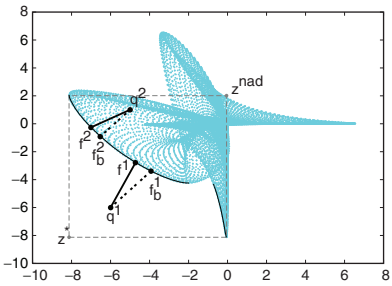


Fig. 2 Preference weights

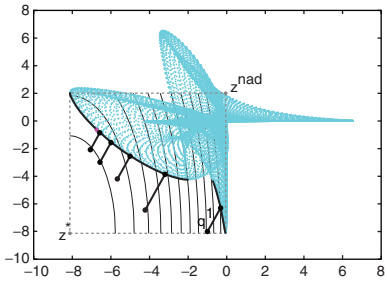
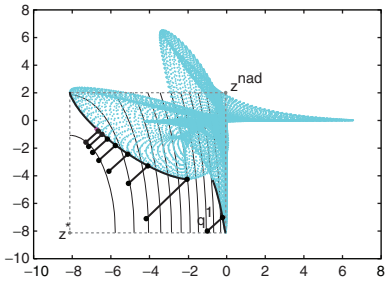


Fig. 3 Basic weights



5 Computational Tests

We have carried out several computational tests in order to compare the performance of our weighting scheme to the solutions generated by using basic weights. With four multiobjective optimization problems described in [12] we have made tests with both real decision makers and with three different types of utility functions (linear, quadratic and exponential). In other words, in the latter type of tests, we have replaced the responses of the DM by utility functions. In each test, we compared the solution of the weighted scheme to the solution obtained with basic weights. We used two settings of tests, Test I and Test II. In the first setting, Test I, we used several single reference points and the solution minimizing the achievement scalarizing function was found. In this test setting, we assumed that our weighting scheme would produce better solutions than basic weights. In the second test setting,

Test II, an interactive solution process was carried out with both weighting schemes and we assumed that the number of solutions that have to be generated before finding the most preferred solution is smaller with our weighting scheme.

With human DMs in Test I, each DM specified different reference points and graded the solution obtained (with achievement functions using our weighting scheme and basic weights) using a scale 1–5 reflecting how well her/his expectations were satisfied (5 indicated that (s)he was very satisfied with the solution obtained). In Test II, DMs again solved each test problem two times but this time using an interactive reference point method incorporating our weighting scheme or basic weights and in each test they tried to find the most preferred solution. They graded the final solutions obtained using the scale 1–5. In addition, we recorded the number of iterations used, that is, how many reference points were needed before the final solution was found. In Table 1, we summarize average values for the grades and numbers of iterations.

With utility functions, we carried out so-called automated tests without involving real DMs. Again, we used two settings, Test I and Test II. In Test I, 100 random reference points $\mathbf{q} = (q_1, \ldots, q_k)^T$ were generated with $q_i \in [z_i^\star, z_i^{\text{nad}}]$ for all $i = 1, \ldots, k$. The gradient of utility function at the generated reference point was used to imitate preference specifications in order to calculate weights for our weighting scheme. This information was used then to solve problem (2). Test I was stopped here whereas in Test II, a new reference point was generated in the gradient direction of the utility function at the solution point obtained, and then a new solution was again computed. The iterative procedure was stopped when some solution had lower utility function value than the solution obtained at the previous iteration. The depicted process was carried out also for basic weights for each test problem and each utility function.

As far as Test I is concerned, our weighting scheme produced a solution with a higher utility than basic weights in 69% of the tests and the average improvement in these cases was 224%.

A summary of the results of automated tests for Test II is given in Table 2. We compared utility function values achieved for the problems CaballeroReyRuiz2 (CRR), ChankongHaimes (CH), PeakFunctions (PF) and modified PeakFunctions (with only two first objective functions - PFM), described in [12]. In Table 2 we list results for each of three utility function types. We have carried out five tests for each utility function and collect here the average results. The first column 'Prob' gives the acronym of the test problem and the second column 'U.f.' the type of the utility function in question. In the third column, 'Max u.' we have the maximal utility function value and in the fourth column 'B.u.' the utility function value of the

Table 1 Tests with human DMs, average values

	Test I (Grades)	Test II (Grades)	Test II (No. of iterations)
Basic	2.75	4.13	6.79
Preference	3.76	4.67	5.56

Table 2 Average results for automated Test II

Prob	U.f.	Max u.	B.u.	P.u.	#(P.u.≥B.u.)	Improve	Impair
	lin.	-128.86	-162.50	-173.86	2	10%	21%
CRR	quad.	43.21	12.71	9.98	4	17%	100%
	exp.	-1304.89	-2028.97	-1648.70	5	29%	–
	lin.	75.46	69.54	70.37	2	5%	1%
CH	quad.	93.51	92.25	93.12	5	1%	–
	exp.	-53.62	-66.85	-61.87	3	20%	7%
	lin.	-1018.95	-1747.89	-1513.33	4	30%	3%
PF	quad.	-687.22	-1215.90	-1236.52	2	16%	15%
	exp.	-297815.99	-14554704.21	-2131364.60	4	5485%	1084%
	lin.	-66.64	-114.71	-107.86	4	26%	54%
PFM	quad.	1.81	0.22	-4.05	0	–	236%
	exp.	-1371.10	-1392.95	-1472.49	1	0%	7%

solution corresponding to basic weights. The next column 'P.u.' shows the utility function value of our weighting scheme. Let us point out that due to the stopping criterion used, the final utility function value does not necessarily have to be close to the maximal utility function value. The three last columns refer to number of test runs (0–5) when our weighting scheme produced a solution with a better utility than the basic one, mean percentage improvement of utility value in cases reported in the previous column and, finally, the mean percentage impairment in the rest of the test runs.

Based on our experiments we can say that our weighting scheme seems to be very intuitive and not very demanding for the DM. The results support our claim that it outperforms the basic weight scheme in a majority of cases. However, the conversion used from an ordinal scale to a cardinal one is rather rough. This means that with more objectives the ratio between different weights is not equal. It would naturally be possible to formulate weights with an equal proportion for all objectives but we want to keep our weighting scheme as simple as possible and, in any case, the results obtained were encouraging.

6 Conclusions

We have suggested a new way of taking preference information coming from the DM more closely into account in interactive reference point based methods developed for multiobjective optimization. Our goal is to be able to produce solutions that are more satisfactory to the DM than the ones produced with standard approaches. In this way, the DM can find the final solution with less iterations. We have also tested our scheme and the results support the usability of our idea. Nevertheless, as has also been reported, the basic weighting scheme may produce better results in some cases. This is why we recommend to use our weighting scheme together with the basic one, and to let the DM choose the best solution.

Acknowledgements This research was partly supported by Teke, the Finnish Funding Agency for Technology and Innovation and its MASI technology programme, Junta de Andalucia (SEJ-445) and Spanish Mobility Program, PR2005-0212.

References

1. Benayoun R, de Montgolfier J, Tergny J, Laritchev O (1971) Linear programming with multiple objective functions: step method (STEM). Math Prog 1:366–375
2. Buchanan JT (1997) A naïve approach for solving MCDM problems: the GUESS method. J Oper Res Soc 48:202–206
3. Jaszkiewicz A, Slowiński R (1999) The 'light beam search' approach–an overview of methodology and applications. Eur J Oper Res 113:300–314
4. Korhonen P, Laakso J (1986) A visual interactive method for solving the multiple criteria problem. Eur J Oper Res 24:277–287
5. Luque M, Miettinen K, Eskelinen P, Ruiz F (2009) Incorporating Preference information in interactive reference point methods for multiobjective optimization. Omega 37: 450–462
6. Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer, Boston
7. Miettinen K, Lotov AV, Kamenev GK, Berezkin VE (2003) Integration of two multiobjective optimization methods for nonlinear problems. Optim Meth Soft 18:63–80
8. Miettinen K, Mäkelä MM (1995) Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. Optimization 34:231–246
9. Miettinen K, Mäkelä MM (1999) Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation. J Oper Res Soc 50:949–959
10. Miettinen K, Mäkelä MM (2002) On scalarizing functions in multiobjective optimization. OR Spectrum 24:193–213
11. Miettinen K, Mäkelä MM (2006) Synchronous approach in interactive multiobjective optimization. Eur J Oper Res 170:909–922
12. Miettinen K, Mäkelä MM, Kaario K (2006) Experiments with classification-based scalarizing functions in interactive multiobjective optimization. Eur J Oper Res 175:931–947
13. Nakayama H, Sawaragi Y (1984) Satisficing Trade-off method for multiobjective programming. In: Grauer M, Wierzbicki AP (eds) Interactive decision analysis. Springer, Berlin, pp 113–122
14. Steuer RE (1986) Multiple criteria optimization: theory, computation, and application. Wiley, New York
15. Wierzbicki AP (1982) A Mathematical basis for satisficing decision making. Math Modell 3:391–405

Interactive Multiobjective Optimization of Superstructure SMB Processes

Jussi Hakanen, Yoshiaki Kawajiri, Lorenz T. Biegler, and Kaisa Miettinen

Abstract We consider multiobjective optimization problems arising from superstructure formulation of Simulated Moving Bed (SMB) processes. SMBs are widely used in many industrial separations of chemical products and they are challenging from the optimization point of view. We employ efficient interactive multiobjective optimization which enables considering several conflicting objectives simultaneously without unnecessary simplifications as have been done in previous studies. The interactive IND-NIMBUS software combined with the IPOPT optimizer is used to solve multiobjective SMB design problems. The promising results of solving a superstructure SMB optimization problem with four objectives demonstrate the usefulness of the approach.

Keywords: Interactive methods · Interior point optimization · IPOPT · Multiobjective optimization · NIMBUS · Simulated moving bed processes · Superstructure

1 Introduction

Real-world optimization problems typically have several conflicting objectives that need to be considered simultaneously. Optimization of real-world industrial processes is often computationally demanding and may require many evaluations of the process model during the optimization. Thus, it is important to have optimization tools that require only few process model simulations in order to obtain an optimal or satisfactory solution quickly.

The problem is too often simplified by optimizing only one objective function, although there usually are several objectives that should be considered at the same

J. Hakanen (✉),
Department of Mathematical Information Technology, University of Jyväskylä,
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
e-mail: jhaka@mit.jyu.fi

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

time. By optimizing several objective functions simultaneously, one can obtain better understanding of the whole problem and its interdependencies. In multiobjective optimization (MOO) [7], we can identify Pareto optimal solutions where no objective can get better values without impairment in at least one of the others. In order to obtain the best solution for the problem in question, we typically need a decision maker (DM), who can express preference information about different Pareto optimal solutions. Some MOO methods try to generate a representation of all Pareto optimal solutions. For many real-world problems these methods can be too slow because they need to generate a large number of solutions. Besides, with more than two objectives, displaying the solutions to the DM is not trivial and it is not easy for the DM to select one of many alternatives as the final solution. Instead, interactive MOO methods aim at decreasing the cognitive burden set on the DM and let the DM to direct the solution process in order to find the most preferred solution [7].

In this study, we consider MOO of Simulated Moving Bed (SMB) processes which have been applied to many important separations in sugar, petrochemical, and pharmaceutical industries. Because they operate dynamically, in periodic cycles, systematic optimization of SMBs remains a challenging problem. Prior to the application of the full discretization approach where both spatial and temporal variables are discretized, optimization of SMB processes was known to be an expensive and challenging problem with long solution times [5].

There exist a number of conflicting objectives associated with these processes including productivity, product quality, utilization of desorbent (solvent) and generation of waste streams. Recently, MOO algorithms have been applied for periodic separation processes for gas separation [6] and for SMB processes [9]. A modified sum of weighted objective functions was used in [6] to obtain a representation of the Pareto optimal set. Note, that the approach is valid only for two objective functions. On the other hand, nondominated sorting genetic algorithm (NSGA) was applied to an SMB problem with two and three objective functions [9]. NSGA tries to approximate the set of Pareto optimal solutions, but it cannot guarantee the Pareto optimality of the solutions obtained.

In order to accelerate the process optimization, an efficient full discretization approach was developed and combined with a large-scale nonlinear programming method for the optimization of SMBs in [5]. More recently, this approach was extended to a superstructure SMB formulation and the ε-constrained method was used to solve the biobjective problem, where throughput and desorbent consumption were optimized [4].

To summarize, we can say that so far, SMB processes have been considered with 1, 2, or 3 objectives, only and all the MOO approaches used have been trying to approximate the whole Pareto optimal set and, thus, have needed to generate lots of Pareto optimal solutions in order to get a good representation. Therefore, they are computationally inefficient. Usually, the ultimate aim is to find a single best solution as the final solution to be implemented and not a set of solutions.

In this paper, we consider a MOO problem arising from the superstructure formulation of SMB introduced in [4]. In our previous study [2], we applied interactive MOO to the standard SMB model and we found that the optimization tools

we used, that is, the interactive design tool IND-NIMBUS based on the NIMBUS method [7, 8] and the interior point optimizer IPOPT [11], were useful in solving the problem. We considered the case with four conflicting objective functions which was a novel way to consider SMBs. Now, we consider the same problem with four objective functions, but using a superstructure formulation instead of the standard one. The superstructure formulation is a more general way to represent SMBs and some novel SMB operating schemes can be obtained [4]. When compared to the standard one, the problem with superstructure formulation has significantly more degrees of freedom and a much richer solution space, but it may require more computational effort to determine optimal solutions.

2 Interactive Process Design Tool

IND-NIMBUS is an implementation of the interactive MOO method NIMBUS [7,8] developed for solving MOO problems arising, for example, in industry. The general idea of the NIMBUS method is to help the DM in finding the most satisfactory compromise between conflicting objectives without generating too many Pareto optimal solutions (see also http://nimbus.it.jyu.fi).

With interactive MOO methods the DM can guide the solution procedure and is able to learn about the behaviour of the problem. (S)he can study the interrelationships of the objective functions and obtain a wider understanding of their effects on the whole problem. Another benefit is that interactive methods are computationally efficient, because only a small number of Pareto optimal solutions usually needs to be computed. On the other hand, interactive methods require the DM to take part in the solution procedure continuously and, thus, the DM has to be willing to devote time to this task. One can expect the DM to find the most preferred solution when (s)he can actively take part in the solution procedure. It is important that the method is easy to use and that the DM can easily answer the questions proposed to her/him.

NIMBUS converts the original multiple objectives together with preference information coming from the DM into a new problem with a single objective function. This new problem can then be solved with appropriate solvers developed for scalar-valued problems. NIMBUS is particularly well suited for problems involving more than two objective functions. IND-NIMBUS has been previously applied to several industrial optimization problems, including the design of papermaking processes and standard SMB processes [1, 2].

We consider the following MOO problem. The vector-valued objective function $f = (f_1, \ldots, f_k)^T$ consists of k real-valued objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ that are to be optimized simultaneously. The decision variables x belong to the feasible region $S \subseteq \mathbb{R}^n$. The objective vectors $z = f(x)$ are in the objective space $\mathbb{R}^k$. We assume that the objective functions are conflicting, that is, all of them do not attain their optima at the same decision vector x. For the simplicity of presentation, we assume that all the objective functions in this section are to be minimized. Thus, the MOO problem considered is of the form

$$\begin{aligned} &\text{minimize } \{f_1(x),\ldots,f_k(x)\} \\ &\text{subject to } x \in S. \end{aligned} \tag{1}$$

Minimization of the vector-valued objective function f is understood in the sense of Pareto optimality: A decision vector $x^* \in S$ is called a Pareto optimal solution if there does not exist another decision vector $x \in S$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1,\ldots,k$ and $f_j(x) < f_j(x^*)$ for at least one j. Provided that the problem is correctly specified, the final solution of a rational DM is always Pareto optimal and, thus, we can consider only Pareto optimal solutions.

The outline of the interactive NIMBUS algorithm is the following. First, a Pareto optimal starting point is generated together with information about the ranges of each objective function in the set of Pareto optimal solutions. We show the ranges to the DM in order to inform her/him about what is possible to achieve in the problem in question.

As a starting point, we use a neutral compromise solution [8], where the average within the range of each objective function is projected to be Pareto optimal. At each iteration, the DM is shown the values of the objective functions in the current Pareto optimal solution $f(x^c)$ and (s)he is asked to indicate how the current solution should be improved. This is done with the help of classifying the objective functions into up to five different classes, that is, the functions f_i whose value i) should be improved as much as possible ($i \in I^{imp}$), ii) should be improved until some specified aspiration level $\bar{z}_i$ ($i \in I^{asp}$), iii) is satisfactory at the moment ($i \in I^{sat}$), iv) can impair up to some specified bound ε_i ($i \in I^{bound}$) and v) can change freely ($i \in I^{free}$).

Note, that $I^{imp} \cup I^{asp} \cup I^{sat} \cup I^{bound} \cup I^{free} = \{1,\ldots,k\}$. The aspiration levels and the upper bounds are given by the DM. In order to be able to produce a new Pareto optimal solution, the classification is acceptable if $I^{imp} \cup I^{asp} \neq \emptyset$ and $I^{bound} \cup I^{free} \neq \emptyset$ due to Pareto optimality. Note, that the definition of the classes does not take into account whether individual objectives are to be minimized or maximized.

According to the classification information provided by the DM, up to four new solutions are generated that try to follow the preferences of the DM as well as possible. In the synchronous NIMBUS method [8], this is realized by forming the corresponding number of new single objective subproblems and the solutions of the subproblems are guaranteed to be Pareto optimal. For details, see [8]. The new problems are solved with a suitable single objective optimizer and in this paper, we use the IPOPT optimizer [11]. The solutions obtained and the current Pareto optimal solution are then shown to the DM, who selects the most preferred one. In IND-NIMBUS, different types of visualizations of the Pareto optimal solutions are provided to the DM in order to aid the comparison and selection. There is also a possibility to generate intermediate Pareto optimal solutions between any two Pareto optimal solutions obtained [8]. Then, if the DM is satisfied with any of the solutions obtained, we terminate the solution procedure. Otherwise the DM can make another classification in the solution (s)he selected.

IPOPT is a large-scale nonlinear optimization package [11] based on a Newton-based interior point (barrier) algorithm with filter line-search method. For a general optimization problem constrained with equality constraints and bounds for the

decision variables, IPOPT applies an interior penalty formulation to convert and solve the problem as a sequence of barrier problems for a decreasing sequence of barrier parameters converging to zero. We assume that the functions in the optimization problem are twice continuously differentiable. It can be shown, that under mild regularity assumptions the solutions of the barrier problems will converge to the solution of the original problem as the barrier parameter goes to zero. For details of IPOPT, see [11] and http://projects.coin-or.org/Ipopt.

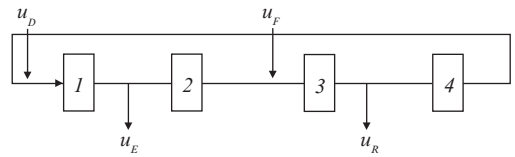
3 Superstructure of Simulated Moving Bed Processes

Modeling of SMB

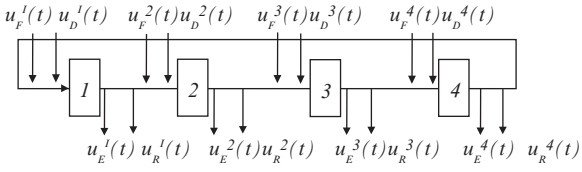
Simulated Moving Bed emerged in the 1960s as a continuous version of chromatographic separation. Since then, SMB has been widely used in pharmaceutical, food, and sugar industries. In particular, separation of chiral isomers has been one of growing application areas in recent years.

An SMB unit consists of multiple chromatographic columns, as shown in Fig. 1(a). Feed and desorbent are supplied into the circulation loop, and extract and raffinate products are withdrawn simultaneously. All of the streams are switched to the adjacent column at a regular interval, or step time. This system does not have a steady state, but has a Cyclic Steady State (CSS), where the concentration profiles repeatedly propagate through the columns.

There have been many nonstandard SMB operating schemes proposed to enhance the performance. For example, VARICOL systems performs asynchronous valve switching, creating larger degrees of freedom [10]. In PowerFeed systems, the liquid velocities are time-dependent. In order to find optimal nonstandard SMB operating schemes, Kawajiri and Biegler [4] proposed an optimization approach



(a) Standard SMB configuration



(b) Superstructure

Fig. 1 Schematic diagrams of four-column standard SMB and SMB superstructure

based on an SMB superstructure. In this approach, the superstructure that embeds numerous kinds of existing and novel operating schemes, as shown in Fig. 1(b), replaces the standard SMB model. Note that the liquid velocities of feed ($u_F^j(t)$), desorbent($u_D^j(t)$), extract($u_E^j(t)$), and raffinate($u_R^j(t)$) are treated as time-dependent variables. In their approach, the optimal operating scheme can be determined by finding the optimal control profiles of the liquid velocities. They also formulated a biobjective optimization problem, with a fixed purity and recovery, and found that the performance of the superstructure can be significantly improved from standard SMBs. In this study, we extend their approach to an interactive multiobjective optimization scheme considering four objective functions: throughput, desorbent consumption, purity, and recovery.

We use the Linear Driving Force model for modeling of chromatographic columns as in our previous study [2]. The throughput $\bar{u}_F$ and desorbent consumption $\bar{u}_D$ are given by:

$$\bar{u}_F := \frac{1}{t_{step}} \sum_{j=1}^{N_{Col}} \int_0^{t_{step}} u_F^j(t)dt, \quad \bar{u}_D := \frac{1}{t_{step}} \sum_{j=1}^{N_{Col}} \int_0^{t_{step}} u_D^j(t)dt,$$

respectively, where t_{step} is the step time. The purity and recovery of the extract product, P_{E_i} and R_{E_i}, are given by:

$$P_{E_i} := \frac{100 \cdot \sum_{j=1}^{N_{Col}} \int_0^{t_{step}} u_E^j(t) C_{E,i}^j(t)dt}{\sum_{j=1}^{N_{Col}} \sum_{i=1}^{Nc} \int_0^{t_{step}} u_E^j(t) C_{E,i}^j(t)dt}, \quad R_{E_i} := \frac{100 \cdot \sum_{j=1}^{N_{Col}} \int_0^{t_{step}} u_E^j(t) C_{E,i}^j(t)dt}{\sum_{j=1}^{N_{Col}} \int_0^{t_{step}} u_F^j(t) C_{F,i}^j(t)dt},$$

respectively, where $C_{E,i}^j(t)$ is the concentration of component i in the extract stream, and $C_{F,i}^j(t)$ is the feed concentration of component i. More information about the superstructure model can be found in [4].

The SMB superstructure model above is fully discretized both in spatial and temporal domains [5]. The spatial domain is discretized via a central difference method. The resulting ODEs are then discretized in the temporal domain using a particular Runge-Kutta discretization: Radau collocation on finite elements.

Multiobjective SMB superstructure

The SMB superstructure problem to be considered is for the separation of fructose and glucose. The values for the parameters in the SMB model are from [3, 5] as given in [2]. The superstructure SMB optimization problem is large having 34 102 decision variables and 34 017 equality constraints. Thus, the superstructure SMB problem has 85 degrees of freedom.

For the MOO of SMB superstructure, we use four different objective functions for this problem: maximize throughput T ($\bar{u}_F$), minimize consumption of solvent, or desorbent D ($\bar{u}_D$), maximize product purity P (P_{E_i}), and maximize recovery of

the valuable component in the product R (R_{E_i}). Thus, our vector-valued objective function is $f = (T, D, P, R)^T$ and the multiobjective SMB superstructure problem is of the form

$$\begin{array}{ll} \text{maximize } \bar{u}_F, \text{ minimize } \bar{u}_D, \text{ maximize } P_{E_i}, \text{ maximize } R_{R_i} \\ \text{subject to} \quad \text{the SMB superstructure model described in [4].} \end{array} \tag{2}$$

These objectives have a number of clear conflicts. For example, if throughput is increased, the feed mixture is more likely to contaminate the products, that is, the purity decreases. Moreover, with uncertain prices and raw material costs, a multiobjective approach can be an extremely useful tool for SMB design.

By imposing upper bounds on objective functions, we eliminate those parts of feasible region that give impractical objective values; it is unlikely that the DM is satisfied with solutions of extremely low throughput, high desorbent consumption, low purity, or low recovery. Restricting the feasible region is also important for numerical stability. For example, if throughput approaches zero, then concentrations of the chemical components in the columns become zero, making the denominators in equations for P_{E_i} and R_{E_i} very small. For the reasons above, we set the following bounds for the objective functions: $T \geq 0.8$, $D \leq 10.0$, $P \geq 90.0$ and $R \geq 85.0$. Note, that the upper bounds for T and R are different from the ones we used in our previous study with the standard SMB [2] (0.4 and 70.0, respectively).

The discretized multiobjective optimization problem (2) is constrained by linear and bilinear equality/inequality constraints. The different objective functions are either linear (T and D) or bilinear (P and R). As a result, the optimization problem is nonconvex.

We use IND-NIMBUS coupled with IPOPT to solve this four objective SMB problem. The SMB superstructure model has been implemented in AMPL modelling language. The multiple objectives are handled by the computationally efficient IND-NIMBUS within a single, generalized NLP formulation, which is solved with the efficient IPOPT optimizer. With this approach, we want to generate new solution candidates efficiently and maintain the interactive nature of the solution procedure.

4 Solution Procedure and Discussion

Next, we describe the solution procedure. A screenshot of the classification window of IND-NIMBUS is shown in Fig. 2. In IND-NIMBUS, the classification is made by clicking different parts of the bars representing the objective functions in the current Pareto optimal solution (single solution in the left in Fig. 2). The value of each objective function is represented with a colored bar that originates from left and right for the objectives to be minimized and maximized, respectively. In both cases, the interpretation is the same: the shorter the colored bars, the better is the corresponding objective function value. All the solutions generated can be seen on the right and interesting solutions can be taken in the set *Best candidates* at any time.

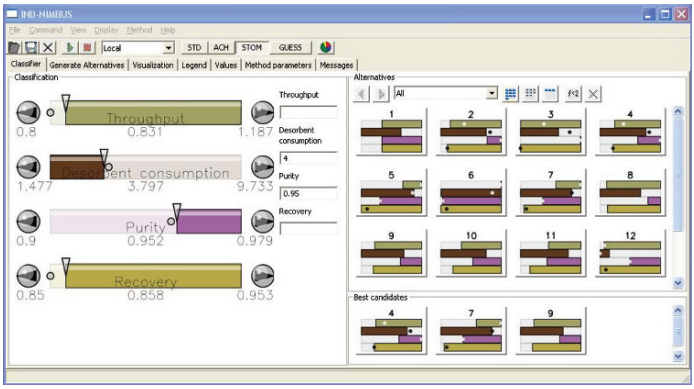


Fig. 2 A screenshot of IND-NIMBUS

The DM involved in this solution procedure was an expert in the field of SMBs. The best and the worst objective function values in the Pareto optimal set were $(1.187, 1.477, 97.9, 95.3)^T$ and $(0.800, 9.733, 90.0, 85.0)^T$, respectively. In other words, those are the approximated ranges for the Pareto optimal set. Note, that we are minimizing desorbent consumption and maximizing all the others, as mentioned previously.

There were altogether four classifications and one generation of intermediate solutions in the solution procedure and they are shown together with the solutions obtained in Table 1. The Pareto optimal solutions preferred by the DM after each action are shown in bold face. As a starting point for MOO, we obtained a neutral compromise solution $z^1 = (0.933, 6.898, 92.7, 88.5)^T$. First, the DM wanted to improve purity and throughput and he was willing to compromise desorbent consumption while the value of recovery was satisfactory. Thus, he made the first classification (I^{imp}={P}, I^{asp}={T}, $\bar{z}_T = 0.95$, I^{bound}={D}, $\varepsilon_D = 1.78$ and I^{sat}={R}) and three different solutions were obtained (z^2, z^3 and z^4).

The DM preferred $z^4 = (0.894, 7.743, 94.9, 87.2)^T$ because it had the highest value for throughput and he chose z^4 as the starting point for the next classification. This time, new Pareto optimal solutions, z^5 to z^7 were obtained (Table 1). In the same manner, Pareto optimal solutions from z^8 through z^{16} were obtained in accordance with classifications, or actions, given by the DM. After these actions, the DM was satisfied with the objective values of $z^{16} = (0.831, 3.797, 95.2, 85.8)^T$, and concluded the solution procedure.

The final solution is shown in both bold face and italics in Table 1. Through the interactive solution procedure, the DM found that 95% purity can be achieved with a moderate increase of desorbent consumption but significantly higher throughput and recovery could be obtained compared to the final solution of our previous study with the standard SMB formulation, $z = (0.461, 1.29, 95.0, 76.2)^T$ [2].

The optimal operating scheme at the final solution of this study is shown in Fig. 3. The figure shows a novel, high performance operation where t_{step} is divided into five, optimally chosen time periods or finite elements. Note, that the feed is introduced

Table 1 Pareto optimal objective function values for superstructure SMB problem

Solution	T [m/h], max	D [m/h], min	P [%], max	R [%], max
Best	1.187	1.477	97.9	95.3
Worst	0.800	9.733	90.0	85.0
z^1	0.933	6.898	92.7	88.5
	$I^{asp}, \bar{z}_T = 0.95$	$I^{bound}, \varepsilon_D = 8.2$	I^{imp}	I^{sat}
z^2	0.860	7.663	96.0	86.1
z^3	0.801	6.683	97.8	85.0
z^4	**0.894**	**7.743**	**94.9**	**87.2**
	I^{imp}	$I^{bound}, \varepsilon_D = 8.5$	I^{sat}	$I^{bound}, \varepsilon_R = 86.0$
z^5	1.066	8.684	92.4	85.0
z^6	1.185	9.74	90.2	85.0
z^7	**1.027**	**8.464**	**92.9**	**85.6**
	4 interm.	solutions between	z^3	z^7
z^8	0.842	7.105	96.3	86.3
z^9	**0.887**	**7.291**	**95.1**	**87.0**
z^{10}	0.932	7.658	94.2	87.0
z^{11}	0.980	8.008	93.5	86.6
	I^{sat}	I^{imp}	I^{sat}	$I^{bound}, \varepsilon_R = 86.0$
z^{12}	**0.827**	**2.742**	**93.9**	**85.0**
z^{13}	0.867	3.383	93.9	85.8
	I^{free}	$I^{bound}, \varepsilon_D = 4.0$	$I^{asp}, \bar{z}_P = 95.0$	I^{free}
z^{14}	0.821	3.554	95.4	85.6
z^{15}	0.800	3.175	95.7	85.0
z^{16}	***0.831***	***3.797***	***95.2***	***85.8***

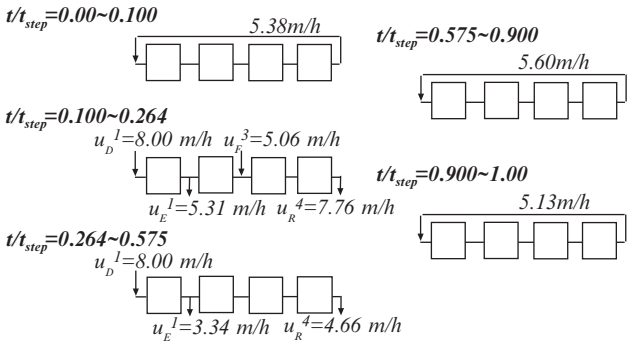


Fig. 3 The optimal operating scheme for the final solution

only in the second finite element, a three zone operation, while only a two zone operation is performed in the second element. On the other hand, the first, fourth and fifth finite elements are characterized by pure recycles with neither input nor output streams. Note, that the optimal structure derived from this approach is very different from the standard SMB shown in Fig. 1(a). Producing a single Pareto optimal

solution took 111 IPOPT iterations (406 objective function evaluations) and 835 CPU seconds on average. (In our previous study with the standard SMB formulation [2] the corresponding numbers were 16.4, 27.6, and 65.8, respectively).

5 Conclusions

For the first time, a four objective optimization problem arising from the superstructure formulation of simulated moving bed processes was succesfully solved. We used the interactive IND-NIMBUS software in MOO where a generalized NLP formulation is solved with the interior point optimizer IPOPT. Previously, we have applied these efficient optimization techniques to solve the four objective problem for the standard SMB model. By using the superstructure formulation we could obtain novel SMB operating schemes and the results show that we were able to obtain a better overall solution when compared to the solution obtained with the standard formulation. The promising results give new perspectives for SMB optimization.

References

1. Hakanen J, Hakala J, Manninen J (2006) An integrated multiobjective design tool for process design. Appl Thermal Eng 26:1393–1399
2. Hakanen J, Kawajiri Y, Miettinen K, Biegler LT (2007) Interactive multi-objective optimization for simulated moving bed processes. Control Cybern 36(2):282–320
3. Hashimoto K, Adachi S, Noujima H, Maruyama H (1983) Models for the separation of glucose/fructose mixture using a simulated moving bed adsorber. J Chem Eng Jpn 16(5):400–406
4. Kawajiri Y, Biegler LT (2006) A nonlinear programming superstructure for optimal dynamic operations of simulated moving bed processes. Ind Eng Chem Res 45(25):8503–8513
5. Kawajiri Y, Biegler LT (2006) Optimization strategies for simulated moving bed and powerfeed processes. AIChE J 52(4):1343–1350
6. Ko D, Moon I (2002) Multiobjective optimization of cyclic adsorption processes. Ind Eng Chem Res 41(1):93–104
7. Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer, Boston
8. Miettinen K, Mäkelä MM (2006) Synchronous Approach in Interactive Multiobjective optimization. Eur J Oper Res 170:909–922
9. Subramani H, Hidajat K, Ray A (2003) Optimization of reactive SMB and VARICOL systems. Comput Chem Eng 27:1883–1901
10. Toumi A, Hanisch F, Engell S (2002) Optimal operation of continuous chromatographic processes: mathematical optimization of the VARICOL process. Ind Eng Chem Res 41:4328–4337
11. Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106:25–57

Scheduling of Water Distribution Systems using a Multiobjective Approach

Amir Nafi, Caty Werey, and Patrick Llerena

Abstract According to technical and economic criteria, a right maintenance policy must be applied to enhance the hydraulic performance and the reliability of water networks. Water networks ensure delivery of water to consumers with the help of pressured networks consisting in several hydraulic components: tanks, reservoirs, pipes, valves, and pumps. The current paper deals with the combination of a multi-objective approach based on Pareto ranking and Genetic Algorithm with a hydraulic software in order to develop a decision support model for pipes renewal. The model is based on the identification of critical pipes and assessment of maintenance costs. It ensures the work planning on pipes and the availability of financial resources at the right time and in the right way. A set of non-dominated solutions according to hydraulic and economic objectives is proposed, selection of the right one remains to the water utility manager. We propose an overview of the developed approach and we present an application on a realistic water network.

Keywords: Decision support model · Genetic algorithm · Hydraulic simulation · Maintenance · Multiobjective · Water network

1 Introduction

Water utilities ensure the delivery of water to the population with the help of network and plants, which allow transport and treatment of water from the source to the end users. We focus on the distribution of water from the source area to the consumers. Historically, the major part of the European water networks was laid after the Second World War until the beginning of the 1980 years. The priorities of water utilities

A. Nafi (✉)
Joint Research Unit Cemagref-Engees in Public Utilities Management, 1 Quai Koch BP 61309, F-67070 Strasbourg, France
e-mail: anafi@engees.u-strasbg.fr

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

are changing. Their goal is not only to deliver water to the end users, but to enhance the reliability of the network by reducing the occurrence of leaks and failures, in order to improve the service level.

Thus, many utilities are dealing with water quality problem, problems water quality, leaks, loss of pressure in pipes, interruptions of water distribution and contamination of water. Those problems can be described by factors dealing with the state of the pipe itself and the environment that brings deterioration. So, water utilities have to prevent occurrence of failures and lack of water by an adequate renewal policy. The objective of our work is to propose a decision tool using failures prediction on pipes by taking into account technical, economic and financial constraints. We have to identify which pipes should be considered? Which rehabilitation should be done? When it should be done? How much will it cost? In which manner the water utility would dispose of sufficient resources to support works on the network? How the budget will be planned? We consider the whole networks by measuring impact of decision about a pipe on pipes remained, the hydraulic deterioration of the network, qualitative variables, assessing and allocating budget to rehabilitation according to a multi-objective approach.

2 Previous Works

Maintenance policy of water networks is widely treated in literature, depending on criteria and approaches used in decision making. We identified methods based on the description of failures occurrences by statistical models that allow to predict damages on pipes using historical data considering past breaks on the network. The different deterioration states of pipe can also be described with the help of a Markov chain. The process is irreversible and the evolution between states is assessed by a transition probability. For these methods the aim is to provide an optimal maintenance policy based on the structural deterioration of pipe. An other approach was proposed by Miettinen [10] which considers a statistical model and objective function handling social costs related to works on network. Kleiner [8] describes the evolution of breaks on pipes in time using the model proposed by Miettinen [10] but they consider more than one intervention on pipes, more rehabilitation alternatives are taken into account. They consider also hydraulic constraints that must be handled in the optimization process. Kleiner [8], Shamir and Howard [12] use Dynamic Programming to propose an optimal scheduling of replacement. Multi-criteria analysis was proposed by Le Gauffre et al. [9] to determine classes of pipes in a network depending on criteria linked to the pipe itself and its environment, the approach uses ELECTRE-TRI method and proposes a classification of pipes giving priority for pipes that must be replaced. Halhal et al. [7] propose an approach based on messy genetic algorithm applied to water networks considering function of hydraulic benefit and technical constraints by a multiobjective optimization. Atiquzzaman et al. [2] propose an application of Non Sorting Genetic Algorithm II (NSGA II) for the design of water network by considering a range of diameter for the pipes in network.

Authors defined two objective functions, the technical assessing pressure benefit assessment and economic for assessing the cost of works on pipes. The proposed application, considered a small network and did not take into account the structural deterioration, the decision taken is based exclusively on the hydraulic operation of the network. Dandy and Engelhardt [3] deals with an optimization model based on genetic algorithm, where the economic criteria consider the deterioration of pipe. Predictions of failures are obtained depending on the pipe material. The model does not consider the reliability of the network. It appears that several methods are adapted to a specific context or number of variables. We propose to take into account more variables that influence the decision of pipes replacement. The main lack concerns the non-consideration of the whole network and more alternatives for renewal works than identical replacement on the network. Few applications consider both physical and hydraulic operation of pipes. The use of Dynamic Programming gives good results at the scale of a pipe and for small network, but it seems unadapted for large networks considering an important set of renewal alternatives. The determination of the different states of pipe is difficult, the distinction of states is not easy to establish and the analysis of the deterioration of buried water pipe is impossible. The use of Markov chain for the description of structural deterioration is more used for inspected infrastructures where we assess the state of the asset according to periodic inspections, specially bridges and roads. We propose in this paper, a hybrid approach based on the use of statistical model, hydraulic simulation ensured by Epanet2 and a multi-objective optimization approach using Modified Genetic Algorithm (MGA) inspired from NSGA II [4]. The model proposes non-dominated solutions according to hydraulic and economic criteria.

3 The Problem Formulation

Water pipes renewal problem concerns both economic and technical objectives. The decision should find a trade-off between the hydraulic performance of the network and costs of works on pipes. In order to involve theses objectives we adopt a multi-objective approach based on Pareto dominance and using genetic algorithm. We define the decision variables and objective functions of the problem as follow.

3.1 Decision Variables

We define a renewal policy *i* on the network as a sequence of interventions (works), that reduce the maintenance cost, water losses on the considered time horizon and enhance the hydraulic performance of the network. The aim is to propose an adequate combination of interventions. We consider three possible alternatives, corresponding to the renewal works on the pipes network: to do nothing and repair if a break occurs, to replace by a similar pipe (same diameter) and to reinforce (enhance diameter). As shown in Fig.1, a renewal policy is a string of alternatives coded by

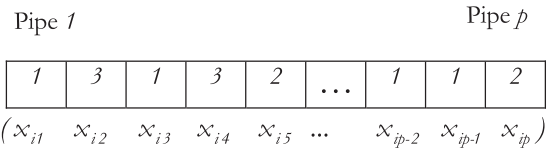


Fig. 1 Decision variables and policy encoding

integers from 1 to 3. Where the code 1 represents the alternative to do nothing. The code 2 represents the alternative to replace and the code 3 the alternative to reinforce. Alternatives represent the design variables of the model. We consider the decision variable x_{ij} as the renewal alternative proposed by the policy i for the pipe j. The length of the string depends on the number of pipes considered in the decision model. For three alternatives and p pipes, we have 3^p possible renewal policies.

3.2 The Objective Functions and Constraints

Genetic algorithms do not use the objective function itself, but consider a fitness function that takes into account the assessment of the solutions according to several objectives. Two objective functions are defined, F_{1i} which assesses the pressure benefit obtained for the policy i and F_{2i} assesses the cost of the renewal works to be done. The constraints considered ensure the right hydraulic performance of the network by the respect of a minimum service pressure P_{min} and maximum pressure P_{max} to avoid an overpressure that can deteriorate pipes. For a given planning horizon Ω, a set of design variables x_{ij}, p pipes involved in the decision making and the defined objective functions, the mathematical formulation of considered problem is given for a policy i as follow:

$$MaximizeF_{1i} = Maximum(Pressure\ benefit) \tag{1}$$

$$MinimizeF_{2i} = Minimum(Renewal\ costs) \tag{2}$$

u.c:

$$P_{min} \leq P(k) \leq P_{max}, k = \overline{1,n}$$

where k is the node index in the network and $P(k)$ is the pressure level of node k. The budget $B(\Omega)_i$ required to achieve a renewal policy i on the network takes into account only the replacement and reinforcement of pipes, the estimation of budget is given by equation below:

$$B(\Omega)_i = \sum_{j=1}^{p} x_{ij}.C_j \ \ and \ \ x_{ij} \neq 1. \tag{3}$$

The objective function F_{1i} measures the pressure benefit available in the network for a given configuration of the network corresponding to a solution i given by the

string of design variables $(x_{i1}x_{i2}...x_{ip})$. For a given solution i and for the pipe j, if the design variable $x_{ij} = 1$, the pipe still unchanged, no modifications are done. If $x_{ij} = 2$, then a replacement of the pipe is required. It corresponds to a modification of the pipes roughness without diameter modification. For $x_{ij} = 3$, the reinforcement is required. The pipes diameter is enhanced and the roughness is modified. For all solutions, modifications corresponding to design variables are carried out, then hydraulic simulations are ensured by Epanet2 to determine the pressure level on consumer nodes. The value of objective function is calculated by:

$$F_{1i}(x_{i1}x_{i2}...x_{ip}) = \sum_{k=1}^{n} \frac{(P(k) - P_{min})}{n}. \tag{4}$$

The objective function F_{2i} assesses the total cost of a given policy, which is expressed as the weighted sum of works cost on pipes, and inverse of hydraulic index HCI_j expressing the hydraulic importance of each pipe j. We assume that costs data are available in water utilities. For the considered network, we calculate the hydraulic critical index for each pipe of the network, after we measure the corresponding cost for each generated policy by the MGA algorithm. For a given solution considered p pipes for renewal with cost C_j for alternativex_{ij} corresponding to the pipe j. Equation (1) assesses the economic objective for i^{th} solution:

$$F_{2i}(x_{i1}x_{i2}...x_{ip}) = \sum_{j=1}^{p} \frac{1}{HCI_j}.x_{ij}.C_j. \tag{5}$$

4 The Decision Support Model

The proposed model takes into account a set of endogenous and exogenous variables that describe the deterioration of pipes. We assess the criticity of pipes network according to future failures and their impact on the hydraulic operation. First we predict a failure rate by analyzing a historic of failures and a set of environment data (nature of soil, length, diameter, installation date) with the help of a statistical model, Proportional Hazard Model described in [10] and [5]. The decision support model involves a hydraulic simulation, which aims to measure the hydraulic importance of each pipe in the network and calculate a Hydraulic Critical Index (HCI). This index expresses the ratio between the water not delivered and the water that should be delivered when a pipe is removed from the network. We propose objective functions that include the importance of each pipe in the network, the structural deterioration process, costs due to works to be done on the network and hydraulic capacity. The optimization process is shown in Fig. 2 .

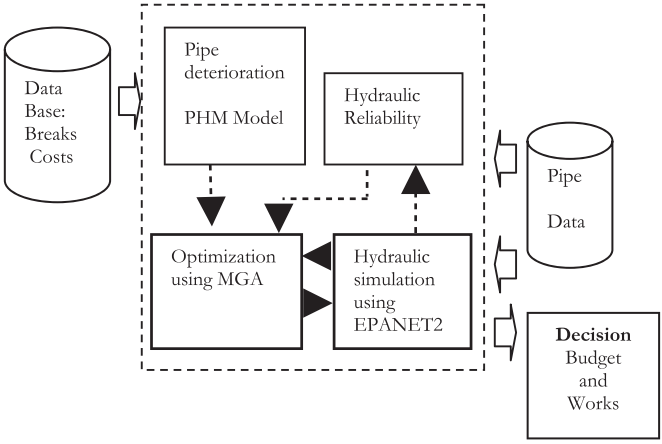


Fig. 2 The decision support model

4.1 Structural Pipes Deterioration

We assume that specific data concerning pipes and renewal costs of works on pipes are available in the water utility. According to the available data, we use a statistical model to describe the structural deterioration process of pipes with the help of statistical functions. We use the PHM developed in [1] and [5] to determine the Survival function S(t). In this process, we have to identify pipes candidates to renewal according to structural deterioration. The selection of these pipes depends on the prediction of future breaks during the planning period, age of pipes and previous failures. Andreou [1] determines three previous failures of pipes as threshold for the acceleration of the deterioration stage. The Survival function $S(t)$ is given by the equation below:

$$S(t) = exp\left[-exp(-\frac{\sum \beta_i . z_i}{\sigma}).t^{\frac{1}{\sigma}}\right] \tag{6}$$

where z_i represents covariates related to environment variables, β_i regression coefficient and σ parameter of scale obtained from available data. Usually the data are observed and checked on a time horizon, called the observation window and noted $[t_a, t_b]$. Where t_a corresponds to the starting year of observation and t_b year of the end of observation. For a pipe j, we assess the occurrence probability of the next break from the calculation of survival function between the date of the last break occurred at time t_l and the date t of occurrence of the next break obtained for a probability of occurrence $P(t)_j$ less than 0.5.

$$P(t)_j = \frac{S(t - t_b)}{S(t_b - t_l)} \tag{7}$$

According to the date of occurrence of the next break, the previous failures and the age of pipes, we select the pipes which are candidates to renewal.

4.2 Hydraulic Reliability

The next step consists in assessing the importance of each pipe in the network operation and measuring the impact of its unavailability. Two indexes are defined. According to the performance reduction after failure occurrences, we identify an index, which expresses the impact of the pipe failure on water supply. We assume that only one pipe failure is possible at the same time (failures are not simultaneous). For each pipe of the network, we simulate a break by closing temporarily the pipe and run a hydraulic simulation using Epanet2. We compare the quantity of water delivered to consumers before Q_{Before} and after Q_{After} unavailability of the considered pipe. The pipe is critical if the amount of water not delivered is important. For a pipe j, Hydraulic Critical Index is given by:

$$HCI_j = \frac{(Q_{Before} - Q_{After})}{Q_{Before}}, j = \overline{1,p} \tag{8}$$

4.3 The Modified Genetic Algorithm: MGA

The previous sub-processes aim to select pipes to be handled in the optimization procedure. According to the nature of the problem, the use of a multiobjective approach is recommended. As defined above, two main objectives are taken into account, the renewal costs and the hydraulic performance of the network. We modify the structure of the Simple Genetic Algorithm by changing the selection procedure. In fact the assessment of solutions is based on a Pareto ranking procedure adapted from [4]. The procedure consists in ranking solutions according to the two objectives considered. At each iteration, the Fitness value of solutions depends on the rank of solution and the hydraulic performance of each policy. In order to ensure a fast convergence of the Algorithm, an elitist approach is applied according to the Non-Sorting Genetic Algorithm II developed by [4]. Each solution i defined by the string $(x_{i1}x_{i2}...x_{ip})$ is assessed according to the objective function F_{1i} and F_{2i} with $x_{ij} = 1,2$ or 3 corresponding to design variables encoding the renewal alternatives considered. At each generation, m represents the number of solutions generated randomly with the help of the genetic algorithm. The ranking procedure identifies non-dominated solutions for a bi-objective optimization problem under constraints. The constraints handled are the minimum pressure required to ensure the proper operation of the network, the maximum pressure not to be exceeded at consumer nodes, we assume that hydraulic model is still unchanged on the planning horizon Ω. All solutions that violate the constraints have a rank less than remain solution's rank.

Fig. 3 The water network studied

For m possible solutions generated, the fitness value of a solution i with the rank r_i is given by:

$$Fitness(i) = \frac{1}{m - r_i + 1}. \tag{9}$$

The assessment of solutions according to objectives defined above is ensured by Epanet2 [13], a ranking procedure and genetic algorithm programmed in VBA Macro on Excel. The decision support model proposed a set of non-dominated solutions.

5 Application

The model was applied on a realistic water distribution network in Alsace (France), after testing on theoretical network. The considered network is composed of 450 pipes, 325 nodes, five tanks and three pumps. The consumption nodes consist in houses, two plants, a swimming pool and an old peoples home. The considered network is shown by Fig. 3.

According to the availability of the data in the water utility, two analysis were carried out. The first concerns the structural deterioration of pipes. We dispose of data on observation window from year 1995 to 2004 related to length, diameter, installation date, nature of soil, soil occupation, traffic level and previous failures. Using the available data, we did a regression in order to assess the parameters of the Survival function S(t) according to PHM model. We identified four significant variables in the deterioration process of pipes: Previous failures (PF), length (L), diameter (D) and age of pipes. The survival function were obtained from the regression on significant variables and defined by the equation (10):

$$S(t) = exp[-26.10^{-5}.(PF+1)^{1.19}.L^{0.717}.D^{-0.902}.Age^{1.167}.t^{1.072}]. \quad (10)$$

The selection of pipes candidates to renewal according to structural deterioration is based on the prediction of the next failures during the planning horizon, the previous failures and the age of the pipes. The analysis of the hydraulic performance indicates that pressure deficiencies appear on several nodes in the peak day (summer day) and peak period, which is situated between 07:00 pm and 09:00 pm.

During the peak demand period, we notice a pressure deficiency in 66 nodes. The solutions proposed by the model must take into account the structural deterioration of pipes and the hydraulic deficiency to enhance network performance. According to hydraulic deficiencies and structural deteriorations, 40 pipes were selected. These pipes are candidate to renewal. The next step is the implementation of the Modified Genetic Algorithm. It allows us to identify the required renewal alternatives for each pipe among those considered into the problem: to repair, to replace or to reinforce. The solutions will be generated by the Genetic Algorithm. For each solution, hydraulic simulation ensured by Epanet2 is done. Feasible solutions must increase pressure deficiencies during the period of peak demand. The pressure in nodes nodes must be between $P_{min} = 20m$ and $P_{max} = 50m$. We implement the MGA on the studied network considering a set of initial population m = 100, number of 100 generations, a crossover probability $P_c = 0.85$ and mutation probability $P_m = 0.015$. Among the 3^{40}possible renewal policies, MGA assesses 10,000 policies.

A set of non-dominated solutions is obtained according to objective functions F_{1i} and F_{2i} and resumed in Table 1. According to solutions shown in the Table 1, MGA proposes a set of non-dominated solutions which are sensitive to available budget and hydraulic performance of the water network. The final decision remains to utility manager depending on available budget and hydraulic performance required. The Fig. 4 shows the Pareto front formed by the non-dominated solutions. After obtaining a set of non-dominated solution, the selection of a solution according to the water utility manager preference may be difficult. The proposed approach should

Table 1 . The non-dominated solutions

(i)	F_{1i}	F_{2i}	$B(\Omega)$	P_{min}	P_{max}	$Renewal$	$Replace$	$Reinforce$
1	20060488,14	9,36	838661,39	48,85	20,90	29	16	13
2	12105676,75	9,30	748444,80	48,86	20,90	22	16	6
3	6520315,06	8,98	446461,55	48,85	20,90	17	12	5
4	8897817,49	9,19	631657,03	48,85	20,90	20	14	6
5	7374055,92	9,06	533807,79	49,14	20,90	18	10	8
6	5729569,19	8,94	403807,40	48,85	20,90	15	11	4
7	5549727,00	8,91	370574,16	48,85	20,90	14	11	3
8	5441391,77	8,74	341987,24	48,85	20,22	12	10	2
9	5127634,44	8,69	304176,53	48,85	20,84	12	8	4

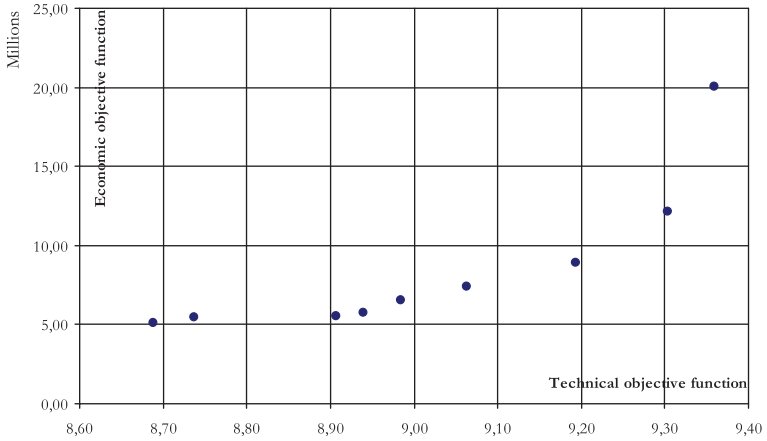


Fig. 4 The Pareto front

be coupled with an appropriate method to help the decision maker to find the most preferred solution among proposed ones.

6 Conclusion

The Modified Genetic Algorithm (MGA) allows us to propose a set of acceptable solutions according to technical constraints, economic and technical objectives. It appears that the model takes into account variables linked to structural deterioration with the help of the Proportional Hazard Model (PHM) Survival function and hydraulic performance. The model should be performed in order to take into account the annual programming of works, detailed annual scheduling is not considered in the current study. More sensitive simulations should be carried out in order to check the proposed solutions, including budget constraints. The approach proposed can be generalized for similar multiobjective problems dealing with structural and operation deterioration of infrastructures disposed on the network. It is clear that specific simulation models are required to describe the operation of infrastructures and to measure the deterioration level. The proposed approach allows trade-offs between economic and technical incommensurable criteria. The trade-off is expressed by the set of possible scenarios assessed according to specific objectives. The uses of a multiobjective approach avoids the definition of weights to express the share of each criterion involved in the decision, in order to reduce the risk of bias. It is also required to help decision maker to select a final solution among non-dominated solutions. Miettinen [10] propose a set of optimization methods that could be coupled with MGA and help water utility manager to select a final solution among Pareto front.

References

1. Andreou SA (1986) Predictive models for pipe break failures and their implications on maintenance strategies for deteriorating water distribution systems. Dissertation, Civil engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
2. Atiquzzaman MD, Liong SY, Yu XY(2006) Alternative decision making in water distribution network with NSGA-II, ASCE. J Water Resour Plan Manage 132(2):122–126
3. Dandy GC, Engelhardt M (2001) Optimal scheduling of water pipe replacement using genetic algorithms, ASCE. J Water Resour Plan Manage 127(4):214–223
4. Deb K, Agrawal S, Pratap A, Meyarivian T (2000) A fast and elitist multi-objective genetic algorithm NSGAII,Technical Report no. 2000001,Indian Institute of Technology, Kanpur, India
5. Eisenbeis P (1994) Modélisation statistique de la prévision des défaillances sur les conduites d'eau potable. Dissertation, Louis Pasteur University, Strasbourg, France
6. Goldberg DE (1994) Algorithmes Génétiques: exploration, optimisation et apprentissage automatique. Addison-Wesley, Paris, France
7. Halhal D, Walters GA, Ouazar D, Savic DA (1997) Water network rehabilitation with structured messy genetic algorithm, ASCE. J Water Resour Plan Manage 123(3):137–146
8. Kleiner Y (1996) Water distribution network rehabilitation: selection and scheduling of pipe rehabilitation alternatives. Dissertation, Dep.Civil.Eng. University of Toronto, Canada
9. Le Gauffre P, Haidar H, Baur R, Poinard D, Schiatti (2004) A multicriteria decision support tool for annual rehabilitation programs of water networks, 4th International Conference on Decision Making in Urban and Civil Engineering Organization, Porto, Portugal
10. Miettinen K (1999) Non-linear multiobjective optimization. Kluwer, Boston
11. Savic D, Walters G (1997) Genetic algorithms for least cost decision of water distribution networks, ASCE. J Water Resour Plan Manage 123(2):67–77
12. Shamir U, Howard CDD (1979) An analytic approach to scheduling pipes replacement. AWWA 71(3):248–258
13. Werey C (2000) Politiques de renouvellement des réseaux d'eau potable. Dissertation, Louis Pasteur University, Strasbourg, France
14. Rossman LA (2001) EPANET users Manual. U.S. Environment Protection Agency, Washington, DC, USA

On Conditional Value-at-Risk Based Goal Programming Portfolio Selection Procedure

Bogumil Kaminski, Marcin Czupryna, and Tomasz Szapiro

Abstract A new goal programming portfolio selection procedure has been proposed in the paper. It uses conditional value at risk at different confidence levels as objectives in a multi-criteria optimisation model. In order to asses the proposed procedure a new comparison method of different portfolio selection models is developed. Based on Warsaw Stock Exchange data it is shown that the proposed approach has better performance than the chosen standard portfolio selection methods.

Keywords: Conditional value at risk · Goal programming · Portfolio selection

1 Introduction

The aim of the paper is to propose a new portfolio selection procedure based on Conditional Value at Risk.

In the portfolio selection problem an investor faces a very large decision space* and conflicting objectives: return maximization and risk minimization. Therefore he or she might have a difficulty in selecting a portfolio that would reflect his or her preferences exactly. The solution of this problem proposed in the literature is to use analytical methods that would help an investor to choose his or her investment portfolio.

Markowitz [7] and then Sharpe [11] proposed to use variance as a risk measure in the portfolio selection problem. Since then other methods for risk measurement

B. Kaminski (✉)
Decision Support and Analysis Division, Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland
e-mail: bkamins@sgh.waw.pl

* The number of assets that can be considered often exceeds 1,000. Such a size of decision space poses a cognitive problem for an investor. For instance it is even not easy to verify that the selected portfolio is non-dominated.

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

have been proposed in the literature and applied in practice. As examples of such approaches one can mention: semivariance approach [3], mean average deviation approach [6], Gini's mean (absolute) difference approach [13], minimal value approach [14], value at risk (VaR) approach [5] and conditional value at risk (CVaR) approach [10]. Each such approach will be called *portfolio selection method* in the paper.

Consider an investor facing a portfolio selection problem. The application of a portfolio selection method to this problem results in some portfolio that is optimal with respect to risk measure used in this method. Such a portfolio will be called a *recommended portfolio*. On the other hand, a portfolio that would be chosen by an investor if he or she followed his or her true preferences will be called an *optimal portfolio*. The difference between optimal portfolio and recommended portfolio will be called *optimality gap*. In the paper we assume that the true preferences of an investor can be described by his or her utility function[†].

Each of the above mentioned portfolio selection methods presents a different measure of an investor's risk aversion. Therefore for each method recommended portfolios and optimality gaps may differ. In the paper we define the *robustness* of the portfolio selection method as its optimality gap against a class investor preferences[‡]. The smaller the optimality gap the more robust the method is.

Value at risk is currently receiving an increasing interest in financial sector. Its use is grounded by recommendations of Basle Committee on Banking Supervision on risk management requirements. However this risk measure has been criticized in the literature. For example [1] shows that it is not coherent – a sum of VaRs of two financial instruments might be lower than VaR of a portfolio consisting of these instruments. Therefore CVaR is proposed in the literature as a risk measure closely related to VaR, but not burdened by its undesired properties.

In the paper we propose a CVaR based goal programming portfolio selection method. The method allows an investor to reveal his risk attitude by specifying CVaR goal levels. This results in a method that is more robust compared to standard approaches. In order to show this we develop a methodology allowing for measurement and comparison of robustness of our method against other methods.

The portfolio selection method taking into account several CVaRs in the recommended portfolio selection has been proposed by Ogryczak [9]. It is based on an interactive multi-criteria approach. However, in practice it is not always possible or desirable to use interactive procedure for portfolio selection. Therefore we restrict the scope of our analysis to non-interactive portfolio selection methods.

The proposed portfolio selection method assumes that an investor chooses several Conditional Value at Risk confidence levels and specifies for each confidence level the desired (goal) CVaR value. We show that this approach ensures that the recommended portfolios are not stochastically dominated (second order)

[†] We assume that an investor behaves as if he or she was making his or her decisions by maximizing some utility function. Analytical form of this utility function does not need be known and in practice it is usually not known.

[‡] The class of investor preferences is described by a set of his or her hypothetical utility functions, see Sect. 4.

and have desirable computational features. Namely the recommended portfolio can be found using linear programming for piecewise linear penalty functions. For non-linear penalty functions genetic algorithm approach is proposed.

We compare recommended portfolios obtained by the method proposed in the paper with recommended portfolios obtained by other portfolio selection methods. The robustness of different methods is measured as the average and maximum difference between recommended portfolio and optimal portfolio for a class of investor's utility functions.

We concentrate on the risk measures that can be interpreted using Yaari's dual utility theory, see [12]. However, the proposed portfolio selection methods comparison approach can as well be used for the class of the risk measures interpretable within von Neumann–Morgenstern utility theory (for instance probability of loss or expected shortfall) or even grounded in a framework that is not based on utility.

The paper is organized as follows. In Sect. 2 we define CVaR and discuss its properties. Next in Sect. 3 we propose the CVaR based goal programming method. Finally in Sect. 4 we compare our method to standard methods found in the literature.

2 Conditional Value at Risk Properties

In this section we define CVaR and discuss its properties that will be used to support the properties of the CVaR goal programming portfolio selection method introduced in Sect. 3.

We define Conditional Value at Risk at a given *confidence level* p as follows:

$$\mathrm{CVaR}_p(X) = \frac{1}{p}\int_0^p F^{-1}(q)dq \tag{1}$$

where random variable X has cumulative distribution function $F(x)$ (see [9]).

One can interpret CVaR, in an intuitive way, as an average over $100p\%$ of worst outcomes in a sample drawn independently from a population with cumulative distribution function $F(x)$ of a random variable X. In particular notice that when $p = 1$ the CVaR reduces to expected value of return.

It is important to note that CVaR criterion is grounded in dual utility theory. Namely for each CVaR level p there exists a dual utility function[§] that gives the same ordering of random variables as CVaR. This is stated in Theorem 1.

Theorem 1. *(see [2]) Consider random variables X_1 and X_2 with cumulative distribution functions $F_1(x)$ and $F_2(x)$ respectively. For dual utility function:*

$$f_p(q) = \begin{cases} \frac{q}{p} & \text{if } q \in [0,p] \\ 1 & \text{if } q \in [p,1] \end{cases}$$

[§] [12] presents the foundations of dual utility theory.

where $p \in [0,1]$ *the following condition holds:*

$$CVaR_p(X_1) \leq CVaR_p(X_2) \Leftrightarrow \int_{-\infty}^{\infty} t df_p(F_1(t)) \leq \int_{-\infty}^{\infty} t df_p(F_2(t)).$$

Notice that the right side of the above equivalence is a comparison of expected values of random variables X_1 and X_2 under dual utility transformation of probabilities.

From the above theorem it follows that there exists such a dual utility function, for which there is no optimality gap for portfolio optimization method based on CVaR at any level p as a goal function (that is: investor's optimal portfolio is exactly the same as recommended portfolio).

Additionally CVaR has also desirable computational properties. It can be shown, see [10], that portfolio optimization method with CVaR as an objective function can be expressed as a linear programming problem by introducing an additional variable v and N nonnegative variables d_i. Parameter p defines confidence level of CVaR and m is the number of assets:

$$\begin{aligned} & v - \frac{1}{pN} \sum_{i=1}^{N} d_i \rightarrow \max \\ & \text{subject to:} \\ & d_i \geq v - R_i x \\ & \forall i \in \{1,\ldots,N\} : d_i \geq 0 \\ & \sum_{i=1}^{m} x_i = 1 \\ & \forall i \in \{1,\ldots,m\} : x_i \geq 0. \end{aligned} \tag{2}$$

By x_i we define the share of asset i value in whole portfolio and as R we define the matrix with N rows and m columns, taking the values of discrete multidimensional random variable describing potential returns on the portfolio[¶].

Based on the definition of CVaR and its properties given by equation 2 in the next section we introduce goal programming CVaR based portfolio selection method.

3 Goal Programming Portfolio Selection Procedure

In this section we define the CVaR based goal programming portfolio selection method. Next we show how it can be solved by linear programming for piecewise linear penalty functions and define genetic algorithm framework used for non-linear penalty functions.

[¶] We assume that for each asset we have N possible returns each occurring with probability $\frac{1}{N}$. In practice the distribution of returns is determined using historical data. Then such random variable is defined by N past observations of returns, see [5].

In the proposed portfolio selection method an investor chooses K CVaRs, their confidence levels p_j and their objective levels g_j. A goal programming problem can be written as:

$$\sum_{j=1}^{K} P\left(g_j - \mathrm{CVaR}_{p_j}(\mathbf{x})\right) \to \min$$

$$\begin{aligned} &\text{subject to:} \\ &\sum_{i=1}^{m} x_i = 1 \\ &\forall i \in \{1,\ldots,m\} : x_i \geq 0 \end{aligned} \tag{3}$$

where $P(\cdot)$ is an increasing penalty function (such an approach allows for penalty functions non symmetric around 0). One can show that optimal solutions of this problem are nondominated:

Theorem 2. *Optimal solution of problem given by equation 3 is nondominated in second order stochastic dominance sense.*

Proof. Notice that function $F(\mathbf{x}) = \sum_{j=1}^{K} P\left(g_j - \mathrm{CVaR}_{p_j}(\mathbf{x})\right)$ is a decreasing function of each $\mathrm{CVaR}_{p_j}(\mathbf{x})$.

Assume that $\mathbf{x}_{opt}$ is an optimal solution of problem given by equation 3. Assume that there exists a non-optimal solution $\mathbf{y}$ that dominates $\mathbf{x}_{opt}$ in second order stochastic dominance sense. Then, using results from [9], for such solution $\forall j \in \{1,\ldots,K\} : \mathrm{CVaR}_{p_j}(\mathbf{x}_{opt}) \leq \mathrm{CVaR}_{p_j}(\mathbf{y})$. But this leads to the conclusion that $F(\mathbf{y}) \leq F(\mathbf{x}_{opt})$ and this contradicts the assumption that $\mathbf{y}$ is non-optimal. □

Theorem 3 shows how the problem given by equation 3 can be expressed in a linear programming form for piecewise linear penalty functions.

Theorem 3. *Assuming* $P(s) = \min(w^+ s, w^- s), w^+ \geq w^-$ *the goal programming problem given by equation 3 can be expressed as:*

$$\sum_{j=1}^{K} w^+ s_j^+ - w^- s_j^- \to \min$$

$$\begin{aligned} &\text{subject to:} \\ &\forall j \in \{1,\ldots,K\} : \left(v_j - \frac{1}{p_j N}\sum_{i=1}^{N} d_i^j\right) + s_j^+ - s_j^- = g_j \\ &\forall i \in \{1,\ldots,N\}, j \in \{1,\ldots,K\} : v_j - r_i x \leq d_i^j, d_i^j \geq 0 \\ &\sum_{i=1}^{m} x_i = 1 \forall i \in \{1,\ldots,N\} : x_i \geq 0 \\ &\forall j \in \{1,\ldots,K\} : s_j^+ \geq 0, s_j^- \geq 0 \end{aligned} \tag{4}$$

Proof. Let $g_j - \mathrm{CVaR}_{p_j}(\mathbf{x}) = s_j^+ - s_j^-$, where $s_j^+, s_j^- \geq 0$. Under this notation we have: $P\left(g_j - \mathrm{CVaR}_{p_j}(\mathbf{x})\right) = w^+ s_j^+ - w^- s_j^-$. Notice that because we are solving a minimization problem either s_j^+ or s_j^- is equal to 0.

Therefore the problem given by equation 3 can be rewritten as:

$$\begin{aligned}
&\sum_{j=1}^{K} w^+ s_j^+ - w^- s_j^- \to \min \\
&\text{subject to:} \\
&\forall j \in \{1,\ldots,K\} : \mathrm{CVaR}_{p_j}(\mathbf{x}) + s_j^+ - s_j^- = g_j \\
&\sum_{i=1}^{m} x_i = 1 \\
&\forall i \in \{1,\ldots,m\} : x_i \geq 0 \forall j \in \{1,\ldots,K\} : s_j^+ \geq 0, s_j^- \geq 0.
\end{aligned} \tag{5}$$

Notice that because either s_j^+ or s_j^- is equal to 0 and $w^+ s_j^+ - w^- s_j^-$ is minimized expression $\mathrm{CVaR}_{p_j}(\mathbf{x}) = s_j^- - s_j^+ + g_j$ is maximized. Therefore we can replace $\mathrm{CVaR}_{p_j}(\mathbf{x})$ using the formula from equation 2. Using this substitution we obtain the optimization problem formulation given by equation 4. □

Although the class of piecewise linear penalty functions is wide it need not always reflect decision makers preferences precisely. Therefore we also consider nonlinear or non convex penalty functions (as for instance it could be assumed that penalty function should not be continuous in 0). For such functions genetic algorithm approach for finding recommended solution is proposed.

In the paper we take the following penalty function as an example[‖]:

$$P(s) = \exp(s) + \alpha[s > 0], \alpha > 0. \tag{6}$$

In such case our optimization problem cannot be linearized and solutions are obtained with genetic algorithm optimization approach.

In the paper we apply the genetic algorithm optimization procedure following [8]. Below we specify: individual encoding scheme, population size, stopping criterion, initial population generation, cross-over and mutation procedures.

We *encode each individual* as a vector of m real numbers representing the share of each asset in the portfolio. For *population size* we use 1,000. The optimization is stopped after 1,000 steps (*stopping criterion*).

In order to *initialize the population* we employ the following procedure:

- For $i \in \{1,\ldots,S\}$, $j \in \{1,\ldots,m\}$ draw $u_{i,j}$ independent random samples from a uniform $0-1$ distribution.
- Normalize the random draws $v_{i,j} = u_{i,j} / \sum_{k=1}^{m} u_{i,k}$.
- Initialize i-th individual as a vector $\mathbf{v}_i = [v_{i,1},\ldots,v_{i,m}]$.

Notice that $\mathbf{v}_i$ meets the constraints of our optimization problem.

In the new population generation we employ the following *cross-over algorithm*:

- Randomly draw two parents $\mathbf{v}_1$ and $\mathbf{v}_2$.

[‖] For logical expression ℓ: $[\ell] = \begin{cases} 1 \text{ if } \ell \text{ is true} \\ 0 \text{ if } \ell \text{ is false} \end{cases}$.

- Generate an m-element random vector $\mathbf{r}$ containing 0 and 1 (each value drawn with probability $\frac{1}{2}$).
- Calculate initial children as:

$$\mathbf{c}_1 = \mathbf{r}\mathbf{v}_1^T + (\mathbf{1}-\mathbf{r})\mathbf{v}_2^T \text{ and } \mathbf{c}_2 = \mathbf{r}\mathbf{v}_2^T + (\mathbf{1}-\mathbf{r})\mathbf{v}_1^T.$$

- Normalize children $\mathbf{v}_i' = \mathbf{c}_i/(\mathbf{1}\mathbf{c}_i^T)$.

Notice that $\mathbf{v}_i'$ meets the constraints of our optimization problem if $\mathbf{v}_1$ and $\mathbf{v}_2$ meet it.

We employ the following *mutation algorithm*:

- Randomly draw a subject $\mathbf{v}$.
- Randomly select its index for mutation. Let it be index t.
- Draw v_t from an uniform $0-1$ distribution (thus update vector $\mathbf{v}$).
- Normalize output by $\mathbf{v}' = \mathbf{v}/\mathbf{1}\mathbf{v}^T$.

Notice that $\mathbf{v}'$ meets the constraints of our optimization problem if $\mathbf{v}$ meets it.

In the next section we compare the linear programming and genetic programming versions of proposed portfolio selection method to standard methods found in the literature.

4 Method Assessment Procedure

In this section propose a procedure for the measurement of robustness of a portfolio selection method. Next using this procedure we evaluate the CVaR based goal programming approach by comparing it to standard methods.

The proposed approach to method robustness assessment is based on optimality gap measurement. In order to measure an optimality gap we assume that an investor has preferences that can be described by dual utility function. A portfolio that maximizes investors expected return under his dual utility function is called *optimal portfolio* (an investor would choose this portfolio if he knew and could apply his dual utility). A portfolio that is chosen by given portfolio selection method is called *recommended portfolio*. The distance between the optimal portfolio and the recommended portfolio is called optimality gap.

In order to evaluate the robustness of different portfolio selection methods the calculation of optimality gap is performed under assumption of different dual utility functions of an investor. Intuitively – the best portfolio selection method should have the lowest optimality gaps for a wide range of investor's dual utility functions.

The approach described above is divided into the following steps:

(a) Select a set of n hypothetical utility functions of an investor and m portfolio selection procedures;

(b) For each utility function i calculate optimal portfolio p_i^o;

(c) For each portfolio selection procedure j calculate recommended portfolio p_j^r;

(d) Calculate deviations $d_{i,j} = \|p_i^o - p_j^r\|$ (distance between an optimal portfolio for i-th hypothetical utility function from portfolio chosen by j-th portfolio selection method-optimality gap);

(e) Compare portfolio selection methods based on average and maximum of square root of mean quadratic deviation of their deviation vectors $[d_{1,j}, \ldots, d_{n,j}]$.

In the paper we employ the following family of investor's hypothetical dual utility functions:

$$f_n(p) = 1 - (1-p)^n, n \in \{2, 3, \ldots, 10\}. \tag{7}$$

We have chosen the following types portfolio selection methods for comparison:

- Non linear EXP, see equation 6, goal programming CVaR ($\alpha = 1$, CVaR confidence levels: 10, 25, 50%);
- LP goal programming CVaR ($w^+ = 2$, $w^- = 1$, CVaR confidence levels: 10, 25, 50%);
- Single objective CVaR maximization (for values of confidence levels p equal to 5, 10, 25, 50, 75 and 100%)**;
- Gini Mean Difference (GMD), see [13];
- Markovitz SD minimization approach (with constraints $E(X) \geq 0$ and $E(X) \geq 1$), see [7].

The goals for the goal programming method were set in the following way. Firstly for postulated dual utility functions we calculated all CVaRs of the optimal portfolio. Secondly we randomly added or subtracted 1% of deviation from the optimal CVaR values (to take into account that desired levels of goals can be evaluated by an investor only approximately before she or he solves the optimization problem).

The comparison of the methods is based on Warsaw Stock Exchange data. We used year 2005 data for 13 stocks included in WIG20 index††.

In Table 1 we present the following comparative statistics of deviation vectors $[d_{1,j}, \ldots, d_{n,j}]$ for each procedure:

- $\mathbf{SqR_{avg}}$: average of square root of mean quadratic deviation;
- $\mathbf{SqR_{max}}$: maximum of square root of mean quadratic deviation.

The goal programming based procedures have best performance for selected reference utility functions. One can also notice that the worst performing portfolio selection method was expected return maximization (CVaR(1.00)).

5 Summary

In the paper we have proposed a new CVaR based goal programming method and developed an approach to assess its performance in reference to other portfolio selection methods. The procedure is grounded in the dual utility theory and gives recommended portfolios that are non dominated in second order stochastic dominance sense, see Theorem 2.

** Notice that CVaR(1.00) reduces to maximization of expected return of the portfolio.

†† WIG20 index includes the biggest companies quoted on Warsaw Stock Exchange. The 13 stocks taken for analysis have been quoted since 01.01.2004 or earlier.

Table 1 Empirical comparison of different portfolio selection procedures for Warsaw Stock Exchange data. GP-based procedures have best average performance

Procedure	$\mathrm{SqR}_{\mathrm{avg}}(\%)$	$\mathrm{SqR}_{\max}(\%)$
GP EXP	3.98	5.52
GP LP	3.69	6.73
CVaR(0.05)	8.32	10.46
CVaR(0.10)	5.08	9.31
CVaR(0.25)	4.81	5.97
CVaR(0.50)	5.02	6.24
CVaR(0.75)	8.42	12.25
CVaR(1.00)	30.80	31.55
GMD	8.01	9.45
Markovitz ($E(X) \geq 0$)	7.12	8.54
Markovitz ($E(X) \geq 1$)	22.87	24.38

We have shown that the method has desirable computational properties. For piecewise linear penalty functions the recommended portfolio can be found by linear programming, see Theorem 3. For non-linear or non-convex penalty function a genetic algorithm framework was proposed.

In the paper we propose to evaluate different portfolio selection methods by comparing optimality gap between investor's optimal portfolio and recommended portfolio. This methodology was applied to compare robustness of CVaR based goal programming method (piecewise linear and non-linear cases) to standard portfolio selection methods found in the literature. The comparison, given in Table 1, shows that the CVaR based goal programming method has the best performance for selected reference utility functions.

The proposed CVaR based goal programming method is applied in the paper within a non-interactive framework. However, it can be extended to allow for an interactive reconstruction of investor's true preferences. This goal can be achieved by allowing an investor to either accept a recommended solution or update the CVaR goal levels and ask for the next recommended solution.

The work presented in the paper focuses on procedures based on dual utility theory. However, the proposed portfolio selection methods comparison approach can be applied for evaluation of other methods giving recommended portfolios against any investor's optimal portfolio derivation approach.

References

1. Artzner P, Delbaen F, Eber JM, Heath D (1997) Thinking coherently. Risk 10:68-71
2. Dhaene J, Goovaerts MJ, Kaas R, Tang Q, Vanduffel S, Vyncke D (2003) Risk measures and optimal portfolio selection. Proceedings of the 7th International Conference on Insurance, Mathematics and Economics. Lyon, France
3. Fishburn P (1977) Mean-risk analysis with risk associated with below target returns. Am Econ Rev 67:116–126

4. Golier C (2001) The economics of risk and time. MIT, Cambridge
5. Jorion P (1997) Value at risk. McGraw-Hill, New York
6. Konno H, Yamazaki H (1991) Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. Manage Sci 37:519–531
7. Markowitz HM (1952) Portfolio selection. J Finance VIII 1:77–91
8. Michalewicz Z (1998) Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, Berlin
9. Ogryczak W (2002) Multiple criteria optimisation and decisions under risk. Control Cybern 31:975-1003
10. Rockafellar R, Urysaev S (2000) Optimisation of conditional value at risk. J Risk 2:21–41
11. Sharpe W (1963) A simplified model for portfolio analysis. Manage Sci 9:277–293
12. Yaari ME (1987) The dual theory of choice under risk. Econometrica 55:95–115
13. Yitzhaki S (1982) Stochastic dominance, mean variance and Gini's mean difference. Am Econ Rev 72:178–185
14. Young M (1998) A minimax portfolio selection rule with linear programming solution. Manage Sci 44:673–683

Optimal Bed Allocation in Hospitals

Xiaodong Li, Patrick Beullens, Dylan Jones, and Mehrdad Tamiz

Abstract In this paper a decision aiding model is introduced for optimizing the allocation of beds in a hospital. The model is based on queueing theory and goal programming (GP). Queueing theory is used to obtain some essential characteristics of access to various departments (or specialities) within the hospital. Results from the queueing models are used to construct a multi-objective decision aiding model in the GP framework, taking account of targets and objectives related to customer service and profits from the hospital manager and all department heads. The model is developed for a public hospital in China. The performance of the model and implications for hospital management are presented.

Keywords: AHP · Bed allocation · Goal programming · Health-care modelling · Queueing theory

1 Introduction

Public hospitals in China are facing more and more economic pressures from the unprecedented development of private hospitals with professional management and high-level medical equipment. Since public hospitals are to be run as profitable institutes of the Chinese health service system, there is increased attention in improving the management of health care resources within public hospitals. In this paper, a decision making model based on the combination of queueing theory and goal programming (GP) is designed to help management assess potential improvements in the allocation of beds to different hospital departments. These models are developed for and tested on the Zichuan Hospital in China.

P. Beullens (✉)
Management Mathematics Group, Department of Mathematics, University of Portsmouth, Portsmouth, PO1 3HF, UK
e-mail: patrick.beullens@port.ac.uk

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

Zichuan Hospital was established in 1949 as a medium-sized hospital, and currently provides a wide range of medical services over eleven departments, including orthopaedics, surgery, nerve medicine, internal medicine, paediatrics, and obstetrics. For reasons of confidentiality, the departments are henceforth identified only by their numeral index. The average occupancy rate of beds is used as a performance measure in the hospital. From 2004 data, an obvious difference in the bed occupancy rates between the departments can be seen. For instance, the average occupancy rate was 98.04 % in department 7, while 47.85 % in department 9. A high rate may indicate revenue loss from patients turned away when no beds are available, and similarly, a low occupancy rate a waste of hospital resources as beds are often left unused.

The general manager of the hospital felt that the problem with bad occupancy rates may have been caused in part by shifts in the arrival patterns of patients for certain specialties due to the increased competition from private hospitals, which often specialize in certain areas. The lack of advanced insights blocked management in making any significant changes. It was decided to build a decision aiding model across the different departments to assess the potential of bed reallocation to improve the hospitals overall performance. The general manager must consider various factors simultaneously, such as the importance of each department, the penalty cost from lost patients, and the holding cost for idle beds. The decision making process should consider not only the overall achievement for the whole hospital but also the performance in each department. Therefore, the decision aiding model must take into account the conflicts between hospital manager and department heads as well as between different departments. Faced with limited resources, a balance needs to be found between the loss from patients turned away and unoccupied beds in each department while being restricted by several constraints, including the fixed total capacity of beds in the hospital.

Various methods from Operational Research (OR) have been developed and applied to deal with problems of health care resource allocation. Research methodologies used include queueing theory (Cooper and Corcoran [2]), simulation (Fetter and Thompson [5]), and mathematical programming. During the last two decades, goal programming (GP) has become more and more a popular approach for allocating resources in health care. Blake and Carter [1] present a methodology including two linear GP models for allocating resources in hospitals to achieve (among other objectives) a preferred mix and volume of types of treatment. Keown and Martin [8] describe the application of integer GP to capital budgeting decisions in hospitals.

Several studies in the literature focus on the problem of bed allocation across different departments, medical specialities, or types of patients in a hospital. Esogbue and Singh [4], for example, develop a mathematical model useful in devising an admission policy in a ward for different types of patients and in which the objective function is made up of shortage and holding costs. Kao and Tung [7] present an approach for periodically reallocating beds to services to minimize the expected overflows. Their demand forecasting system uses an $M/G/\infty$ queueing model to approximate the patient population dynamics for each service. Their study

emphasizes that forecasts should form the basis for analysis and accepts approximation by queueing models for the sake of procedural simplicity. Lapierre et al. [10] consider the problem of allocating a number of beds to different medical and surgical specialities in a hospital, taking into account that the scheduling of medical procedures varies over a week and that the demand for various medical services shows seasonality. A time series model is developed by using hourly census data to make good decisions regarding the size of each unit. Similarly, to determine the frequency distribution associated with a hospital care units census, Cote [3] present an analytic approach based on a modified version of the Holt-Winters multiplicative seasonality forecasting model.

Very few studies, however, consider the bed allocation problem as a multi-objective problem. Kim et al. [9] demonstrate with a validated computer-simulation model that there is no single dominant solution to the bed-allocation problem, thus indicating the multi-objective nature of the problem. Oddoye et al. [12] detail a GP model to analyze the performance of a Medical Assessment Unit (MAU) and find solutions for bed allocation for patients with minimum delay. However, due to the deterministic nature of the model built, it is difficult to investigate the systems performance for optimal work flow. Hence, Oddoye et al. [11] present a simulation model considering the factors in length of stay, number of beds, nurses and doctors in the MAU. Thereafter a GP model is applied to perform trade-off analysis, using the results from the simulation model. The hourly allocation schemes of resources can be deployed within the MAU to help minimizing delays and increase the flow of patients.

This paper contributes to the research literature on multi-objective decision making for bed allocation by introducing a novel mathematical model which combines queueing theory and GP. It is assumed that every department (or speciality, or type of treatments needing similar care) is to be awarded a fixed number of beds. It is assumed that the number of beds affects the performance of a department in terms of (1) the acceptance probability when a new patient arrives and (2) the average daily profits. The queueing model of Gorunescu et al. [6] is applied (Sect. 2) to find for each department, respectively, the number of beds required to achieve a targeted acceptance probability, and the number of beds needed to optimize daily profits. Results show that these objectives are conflicting. Hence, the queueing model is used to model (1) and (2), respectively, as functions of the number of beds awarded. Thereafter, a GP model is formulated using the queueing model results and solved to obtain an allocation of beds to each department subject to a limit on the total number of available beds for the hospital (Sect. 3). Case study results using 2004 data from the Zichuan Hospital in China illustrate how bed reallocation affects performance, according to the predictions of this model, of each department and the hospital in general (Sect. 4). Finally, conclusions and recommendations for future research are given (Sect. 5).

2 Application of Queueing Models

Queueing models are widely used in industry and business to improve customer services, such as in the airlines and telecommunication sectors, in banks, and supermarkets. These models are particularly useful in determining capacity needs in the context of stochastic arrival patterns and service times. In the context of hospital bed allocation, Gorunescu et al. [6] argue that patient arrivals for a specialty or department can be fairly approximated by a Poisson process, and patient service time following a phasetype distribution. They hence develop an $M/PH/m$ queueing model where the number of beds available is fixed to m and no queuing is allowed; a patient finding that all m beds are occupied, is lost. If the Poisson arrival and steady state distributions hold, this $M/PH/m$ queuing model is valid. Previous studies suggest the appropriateness of this model, even when there is some seasonality in patient admissions. With a Poisson arrival rate of λ per unit time (day) and an average length of stay of τ days, the average number of arrivals during an average length of stay is therefore $\lambda\tau$. Assuming the queueing system is in steady state, the probability L that some arrivals are lost because all m beds are occupied can be given by (1) and then the probability of admitted patients can be represented by (2), where k is defined as the number of phases of the service time [6]:

$$L = \frac{(\lambda\tau)^m/m!}{\sum_{k=0}^{m}\left[(\lambda\tau)^{(k)}/k!\right]}, \tag{1}$$

$$c^a = 1 - L. \tag{2}$$

In order to measure the utilization of beds in a speciality, they defined the bed occupancy rate ρ by:

$$\rho = \lambda\tau(1-L)/m. \tag{3}$$

From this formula, the probability of admitted patients for different values of number of beds m can be calculated. It is found to be a concave function which asymptotically approaches 100 % as illustrated in Fig. 1.

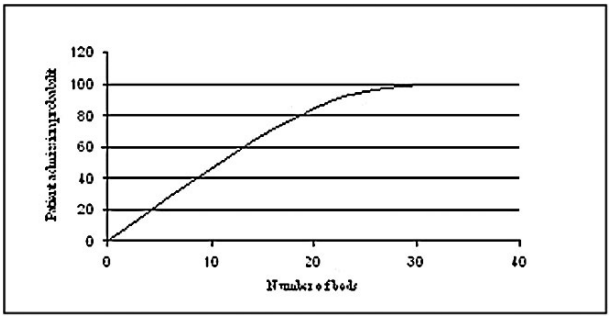


Fig. 1 Patient admission probability as a function of the number of beds (Department 1)

In addition, Gorunescu et al. [6] introduced a base stock inventory model to find the relationship between the average profit per day and the number of beds. The three components that result in total profit in this model are related to: the revenue r (minus direct treatment costs) per day generated from each admitted patient; the penalty cost as a fixed percentage of the lost revenue for each patient turned away; and the holding cost h per day and per idle bed. According to queuing theory, the average number of patients in the hospital equals $\lambda\tau(1-L)$; the average demand per unit time that is lost equals λL; and the average idle-bed inventory equals $[m-\lambda\tau(1-L)]$. Therefore, the average profit per day c^p for each department is:

$$c^p = r\lambda\tau(1-L) - \pi L - h\left[m-\lambda\tau(1-L)\right]. \tag{4}$$

The values of average profit per day corresponding to different number of beds can therefore be calculated. See Fig. 2. The optimal profit per day is achieved by balancing the number of delayed patients against the number of empty beds. As seen in Fig. 2, the profit per day is a concave function starting from a negative value when the number of beds approaches zero. As the number of beds increases, more patients can be admitted generating more revenue and lowering penalty costs. Increasing the number of beds above the optimal number decreases profits due to the excessive holding costs of idle beds.

Both c^a as a measure of the level of service offered, and c^p as a measure of profits obtained, are seen as considerably important criteria, and more useful concepts as opposed to bed utilization rates, to judge whether an allocation of hospital resources is reasonable.

The above two models were used to obtain the values of patient admission probability and profit per day with various numbers of beds for each department in the hospital. Taking 95% as an ideal value for admission probability, and the optimal profit per day as a second ideal value for profit, the optimal number of beds for each department are as summarized in Table 1.

The total number of beds needed to get the ideal target of patient admission probability is 280, which is, purely by accident, exactly equal to the current amount of beds in the hospital. If all departments would need to be satisfied in getting their

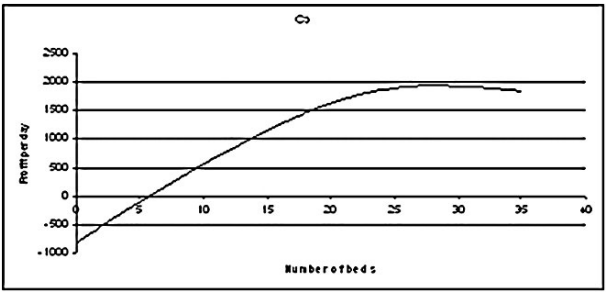


Fig. 2 Profit per day as a function of the number of beds (Department 2)

Table 1 Optimal number of beds to reach the ideal values for each department

Department	Number of beds in respect to 95% of patient admission	Number of beds in respect to optimal profit per day
1	25	29
2	25	28
3	30	34
4	31	36
5	26	30
6	40	46
7	49	58
8	25	26
9	10	11
10	6	6
11	13	15
Total	280	319

own optimal profit per day, however, the number of beds required is 319, which exceeds the existing 280. The limit of 280 beds is viewed as a hard constraint by the hospital manager.

3 GP Model Description

Table 1 shows that the existing 280 beds in the hospital are not able to satisfy all departments to reach their targets. An optimal bed allocation for the hospital manager therefore will have to violate some departments individual objectives. It is thus important to consider when distributing the 280 available beds among the departments the trade-offs between objectives in patient admission and profit performance, as well as among all decision makers including the hospital manager and every department head. The objective of the GP model is to offer a decision aiding tool that can help making such trade-offs.

3.1 Assumptions

Objectives are produced from all the decision makers including the hospital manager and department heads. They try to achieve two types of objectives: patient admission and profit per day. These two objectives are conflicting in the sense that an optimal allocation of resources for one objective is in general not optimal for the other. For each objective, a two-level target (lower and upper) is defined based on the concept of a penalty function in goal programming [13]. The flexibility as such introduced

in the model allows certain departments to concentrate on admitting more patients since their patients may need urgent treatment. Other departments, however, having a work flow pattern of generally less critical services may wish to concentrate more on how to increase profits per day. Note that 'department' in this model may be loosely interpreted; it may also refer to a speciality or any grouping of patients requiring similar treatment.

From Figs. 1 and 2, it can be seen that piecewise linear approximations offer an effective way to represent these concave functions in the GP model. A mixed integer linear programming methodology can then be used, introducing breakpoints in the expression of the GP model constraints:

b_i^a = number of breakpoints in patient admission probability function of department i.

b_i^p = number of breakpoints in profit per day function of department i.

The following notation is used for the (integer) decision variables:

x_{ij} = number of beds between breakpoint $(j-1)$ and j of admitting patient in department i.

y_{ij} = number of beds between breakpoint $(j-1)$ and j of profit per day in department i.

The following parameters are also introduced:

c_{ij}^a = unit value of patient admission probability for x_{ij}.

c_{ij}^p = unit value of profit per day for y_{ij}.

In addition, target values and deviations in the GP model are represented as follows:

g_i^Q = target values of different objectives for decision maker i.

n_i^Q = negative deviations associated with the targets of the objectives for decision maker i.

p_i^Q = positive deviations associated with the targets of the objectives for decision maker i,

where $Q = a1, a2, p1, p2$; and $i = 0, 1, \ldots, 11$; $a1$ and $a2$ represent the targets at the upper and lower-level of patient admission probability; and $p1$ and $p2$ imply those at the upper and lower-level of profit per day, respectively. Parameters and variables with index $i = 0$ are from the hospital manager, and $i = 1, \ldots, 11$ represents the department number.

3.2 Objective Function

The aim of the GP model is to minimize all the negative deviations from patient admission probability and profit per day. The weighted GP technique is used with the following achievement function:

$$\text{Min} = \sum_{i=0}^{11} \left(\frac{w_i^{a1}}{g_i^{a1}} n_i^{a1} + \frac{w_i^{a2}}{g_i^{a2}} n_i^{a2} + \frac{w_i^{p1}}{g_i^{p1}} n_i^{p1} + \frac{w_i^{p2}}{g_i^{p2}} n_i^{p2} \right), \tag{5}$$

where w values are the weights derived from the preferences of decision makers by the AHP technique [14]. In order to avoid arbitrary solutions which do not correspond to the actual preference of the decision makers, a percentage normalizing procedure is implemented without introducing new deviational variables [15].

3.3 Model Constraints

The two-level target values of each objective of decision makers are reflected by (6) – (13). The total number of beds is a hard limit in this model ((14) and (15)). Equation (16) expresses that the number of beds chosen in both objectives for each department must be equal. Equations (17) and (18) add the restriction that the value of the variable cannot exceed the number of beds ($\bar{x_{ij}}$ and $\bar{y_{ij}}$) between breakpoint $(j-1)$ and j.

$$\sum_{j=1}^{b_i^a} c_{ij}^a x_{ij} + n_i^{a1} + n_i^{a2} - p_i^{a1} = g_i^{a1} \quad i = 1, \ldots, 11, \tag{6}$$

$$\sum_{j=1}^{b_i^a} c_{ij}^a x_{ij} + n_i^{a2} - p_i^{a2} = g_i^{a2} \quad i = 1, \ldots, 11, \tag{7}$$

$$\left(\sum_{i=1}^{11} \lambda_i\right)^{-1} \left[\sum_{i=1}^{11} \lambda_i \left(\sum_{j=1}^{b_i^a} c_{ij}^a x_{ij}\right)\right] + n_o^{a1} + n_0^{a2} - p_0^{a1} = g_0^{a1}, \tag{8}$$

$$\left(\sum_{i=1}^{11} \lambda_i\right)^{-1} \left[\sum_{i=1}^{11} \lambda_i \left(\sum_{j=1}^{b_i^a} c_{ij}^a x_{ij}\right)\right] + n_0^{a2} - p_0^{a2} = g_0^{a2}, \tag{9}$$

$$\sum_{j=1}^{b_i^p} c_{ij}^p y_{ij} + n_i^{p1} + n_i^{p2} - p_i^{p1} = g_i^{p1} \quad i = 1, \ldots, 11, \tag{10}$$

$$\sum_{j=1}^{b_i^p} c_{ij}^p y_{ij} + n_i^{p2} - p_i^{p2} = g_i^{p2} \quad i = 1, \ldots, 11, \tag{11}$$

$$\sum_{i=1}^{11} \sum_{j=1}^{b_i^p} c_{ij}^p y_{ij} + n_0^{p1} + n_o^{p2} - p_0^{p1} = g_o^{p1}, \tag{12}$$

$$\sum_{i=1}^{11} \sum_{j=1}^{b_i^p} c_{ij}^p y_{ij} + n_0^{p2} - p_0^{p2} = g_o^{p2}, \tag{13}$$

$$\sum_{i=1}^{11} \sum_{j=1}^{b_i^p} y_{ij} \leq B, \tag{14}$$

$$\sum_{i=1}^{11} \sum_{j=1}^{b_i^p} x_{ij} \leq B, \tag{15}$$

$$\sum_{j=1}^{b_i^a} x_{ij} = \sum_{j=1}^{b_i^p} y_{ij}, \tag{16}$$

$$x_{ij} \leq \bar{x}_{ij} \quad i = 1, \ldots, 11; j = 1, \ldots, b_i^a, \tag{17}$$

$$y_{ij} \leq \bar{y}_{ij} \quad i = 1, \ldots, 11; j = 1, \ldots, b_i^p. \tag{18}$$

$n_i^Q, p_i^Q \geq 0$, and x_{ij}, y_{ij} are non-negative and integer for all i and j.

In (8) and (9), the patient admission probability for the hospital as a whole is determined as the average admission probability where the weights in the average are given by the different arrival rates. In (12) and (13), which specify the profit targets for the hospital as a whole, the hospital manager cannot assume the sum of the target of each department head as his/her own profit target value because the total number of beds is restricted. Thus, an additional model for determining the maximal feasible value is needed. This model is to be solved before the actual GP model, and has the following objective function:

$$\text{Max} g_0^{p1} = \sum_{i=1}^{11} \sum_{j=1}^{11} c_{ij}^p y_{ij} \tag{19}$$

subject to the set of (10), (11), (14), (18) and $y_{ij} \geq 0$ and integer. The outcome of this model sets the upper level target for the general manager in (12).

The GP model presented keeps the total number of beds constant. It can of course be easily modified to other values. In practise, it may be that certain departments are limited in space for beds or cannot accept large changes in beds compared to the current situation for other reasons. This concern may arise, for example, from envisaged limitations for changing the capacity of other resources such as special nursing capabilities or specialists. To model such concerns, the specification of additional lower and upper bounds on the number of beds that may be awarded to a department can be easily introduced as additional constraints. Such constraints were not introduced for the case study presented in Sect. 4.

4 Discussion of Results

Due to limitations of space, the results of the application of the GP model to the Zichuan Hospital data of 2004 are only briefly described. The current and new numbers of beds per department, as predicted by the model, are displayed in Fig. 3. Note that Department 7 will obtain twelve more beds in order to satisfy the requirement of a large amount of patients. The other departments, however, are much less affected.

4.1 Patient Admission

The change in performance of patient admission probability as predicted by the GP model is displayed in Fig. 4. Seven departments cannot achieve the upper-level

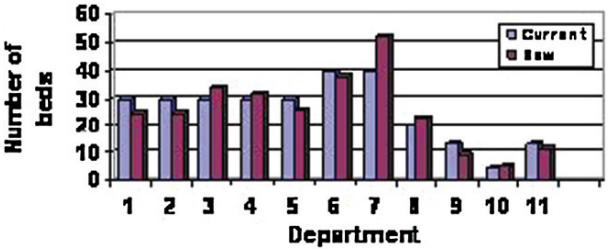


Fig. 3 Current allocation of beds and new allocation found with GP

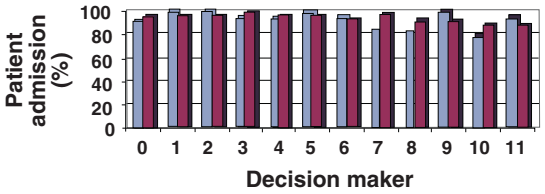


Fig. 4 Current and new values of patient admission probability

target of 95% while only departments 1, 3, 4, and 7 can admit more than 95% of coming patients in the new situation. The results further predict that five departments will improve their performance, especially department 7. There is also a significant increase from 90.59 to 94.94% in the performance of the entire hospital.

4.2 Profit per Day

The results of the model for daily profits are displayed in Fig. 5. As the upper-level targets of profit are defined as those reached with an optimum number of beds for each department, and since the total number of beds needed for this to happen largely exceeds the available number of beds (see Table 1), it was to be expected that the majority of departments are not able to achieve their upper-level target in the new bed allocation. Department 3 is the only speciality that reaches its top target in the new configuration. There are six departments which produce a higher profit, with department 7 again achieving the largest improvement. Most departments reach at least 90% of their optimum. The hospitals overall achievement with 280 beds is now almost at its perfect level; it increases its profits from 93.36 to 99.64% of the optimum.

4.3 Bed Occupancy

It is interesting to also review how the bed reallocation suggested by the model alters the bed occupancy rate per department as this is a performance measure used in

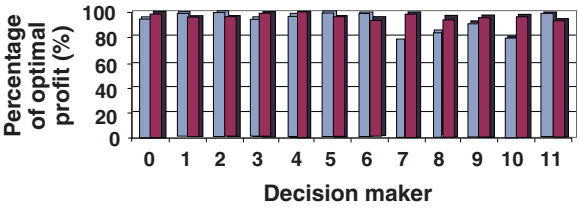


Fig. 5 Current and new values of profits as a percentage of the maximum achievable profit in the queueing model

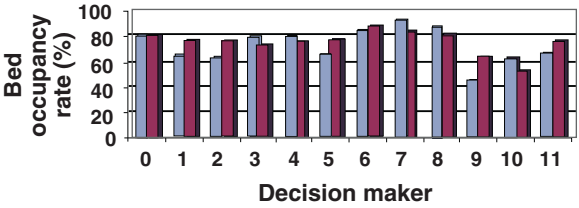


Fig. 6 Current and new values of bed occupancy rates

many hospitals. Results are given in Fig. 6. The hospitals total bed occupancy is lowered, but the performance in patient admission probability and profits is improved. While all departments who see their bed occupancy rates fall gain more beds and a higher patient admission probability, and vice versa, the impact on profits is less predictable. With the sole exception of department 9, however, all departments who decrease (increase) their occupancy rates increase (decrease) their profits. These findings perfectly illustrate that high bed occupancy rates are actually often not desirable for both patient admission probability and profits.

Generally, between 75 and 80% of the beds in most departments are occupied in the new bed allocation which is seen as an acceptable level in practice. Note, however, that department 10 actually decreases its occupancy rate to 50% while it improves patient admission probability and profits. Not conforming to the industry practice of an occupancy rate around 80% therefore does not necessarily imply bad performance. Overall, these comparisons make clear that bed occupancy rates are not adequate performance measures and should not be used by themselves to evaluate the performance of a department or to steer bed reallocation policies in hospitals.

5 Conclusions

This paper is concerned with the strategic problem in a hospital of bed reallocation among its departments. A decision making model is built based on a combination of queuing theory and goal programming. The $M/PH/m$ queuing model can be used

effectively to investigate some essential characteristics of access to service in the hospital. It provides a means of calculating patient admission probability and profit achievement as functions of the number of beds for each individual department. The basic assumptions for it to be applicable are that the arrival process is Poisson and that results reflect a steady state situation. While this queuing model can be criticized for lacking the modelling power of other approaches such as discrete event simulation, it does lead to results based on the simple measurement of average arrival rates and average residence times for each department, without further requirements of detailed modelling. The Goal Programming methodology that uses the results of the queuing model as input, however, does not exclude the use of other techniques for providing this input.

The GP methodology has been used to construct a multi-objective decision aiding model taking account of targets and objectives of the hospital manager and the department heads. The case study results show that the reallocation of beds leads the hospital to overall larger profits and a higher admission probability. While some departments will gain in performance, other departments may have to accept a lower performance. In this case study, this has not resulted in large conflicts as departments with a bad performance largely benefit from the reallocation while excellent performance departments see their performance only slightly decreased. In comparison to the presented approach, the use of bed occupancy rates has been shown to be ill-equipped for estimating the performance of a hospital department in terms of customer service and profits.

Bed allocation in this paper was made to departments. It is highly likely that more complex policies such as pooling of beds between departments may further lead to increased performance. This forms an interesting area for further research. The approach presented may serve as a starting point for the development of capacity planning models for businesses operating in similar conditions, such as the allocation of beds or room types in hotels.

References

1. Blake JT, Carter MW (2002) A goal programming approach to strategic resource allocation in acute care hospital. Eur J Oper Res 140(3):541–561
2. Cooper JK, Corcoran TM (1974) Estimating bed needs by means of queueing theory. New Engl J Med 291(22):404–405
3. Cote MJ (2005) A note on 'bed allocation techniques based on census data'. Socioecon Plan Sci 39:183-192
4. Esogbue AO, Singh AJ (1976) A stochastic model for an optimal priority bed distribution problem in a hospital ward. Oper Res 24(5):884–898
5. Fetter RB, Thompson JD (1965) The simulation of hospital systems. Oper Res 13(5):689–711
6. Gorunescu S, McClean SI, Millard PH (2001) A queueing model for bed-occupancy management and planning of hospitals. J Oper Res Soc 53:19–24
7. Kao PC, Tung G (1981) Bed allocation in a public health care delivery system. Manage Sci 27(5):507–520
8. Keown AJ, Martin JD (1976) An integer GP model for capital budgeting in hospitals. Financial Manage 5(3):28–35

9. Kim SC, Horowitz I, Young KK, Buckley TA (2000) Flexible bed allocation and performance in the intensive care unit. J Oper Manag 18: 427–433
10. Lapierre SD, Goldsman D, Cochran R, DuBow J (1999) Bed allocation techniques based on census data. Socioecon Plan Sci 33:25–38
11. Oddoye JP, Tamiz M, Jones DF, Schmidt P (2007) A simulation model for health planning in a medical assessment unit with multi-objective output analysis. Technical report, Management Mathematics Group, University of Portsmouth, UK
12. Oddoye JP, Yaghoobi MA, Tamiz M, Jones DF, Schmidt P (2007) A multi-objective model determine efficient resource levels in a medical assessment unit. J Oper Res Soc
13. Romero C (1991) Handbook of critical issues in goal programming, 1st edn. Pergamon, Oxford
14. Saaty TL (1994) How to make a decision: the analytic hierarchy process. Interfaces 24(6): 19-43
15. Tamiz M, Jones DF, Romero C (1998) Goal programming for decision making: an overview if the current state-of-the-art. Eur J Oper Res 111:567-581

Multiobjective (Combinatorial) Optimisation – Some Thoughts on Applications

Matthias Ehrgott

Abstract In recent years there have been considerable advances in methodology (exact and heuristic algorithms) to solve multiobjective optimization problems. Combined with the rapid improvement in computing technology, this means that large scale multiobjective optimization problems arising in real world applications have become tractable.

In this paper, I outline some application areas and illustrate how the application of multiple objective methods provide secondary benefits such as additional insight in the application area and improved processes. These benefits are in addition to the primary benefits of more realistic modelling and better decision support by including the conflicting goals that decision makers usually face.

However, the study of real world applications also motivates research on the mathematical aspects of multiobjective optimization. I illustrate this win–win situation using examples drawn from finance (portfolio optimization), transportation (train timetabling, airline crew scheduling), medicine (radiotherapy treatment planning), and telecommunication (routing in networks).

Keywords: Multiobjective optimization · Optimization in medicine · Real world application · Transportation

1 Mathematical Formulation

A multiobjective optimisation problem is the following mathematical programme

M. Ehrgott
Department of Engineering Science, The University of Auckland, Private Bag 92019, Auckland Mail Centre, Auckland 1142, New Zealand
e-mail: m.ehrgott@auckland.ac.nz

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

$$\begin{aligned} \min f(x) &= (f_1(x), \ldots, f_p(x)) \\ \text{subject to } g(x) &\leqq 0, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}^p$ is a vector-valued objective function and $g(x) : \mathbb{R}^n \to \mathbb{R}^m$.

In the case of integer programmes, that I am mainly concerned with in this paper, I further assume that f and g are linear functions. Thus a multiobjective integer programme is

$$\begin{aligned} \min f(x) &= Cx \\ \text{subject to } Ax &\leqq b \\ x &\in \{0,1\}^n. \end{aligned}$$

I denote by $X = \{x \in \mathbb{R}^n : g(x) \leqq 0\}$ or $X = \{x \in \{0,1\}^n : Ax \leqq b\}$ the feasible set in decision space and by $Y = f(X) = \{f(x) : x \in X\}$ the feasible set in objective space. I understand solving a multiobjective integer programme as finding a complete set of efficient solutions X_E, according to the definition of [11]. A feasible solution $\hat{x}$ is efficient if there is no $x \in X$ with $f(x) \leqq f(\hat{x})$ and $f(x) \neq f(\hat{x})$. The image of the efficient set in objective space is the set of non-dominated points $Y_N := f(X_E)$.

It is of course impossible to give a comprehensive survey of applications of multiobjective optimisation in the space of this paper. I have therefore made a very subjective selection of problems that I am familiar with. They are nevertheless drawn from widely different application areas. In each of the examples I emphasise why I find it instructive and what lessons can be learned. The applications I consider are the portfolio selection problem in finance (Sect. 2), train timetable information and airline crew scheduling from the transportation field (Sects. 3 and 4), radiotherapy treatment design in medicine (Sect. 5) and the telecommunication application of routing in IP networks (Sect. 6).

2 Finance: Portfolio Optimisation

The first problem I want to discuss is the portfolio optimisation problem of deciding on an investment of a certain sum of money, for example at a stock exchange, so as to maximise the return and minimise the associated risk. If the number of websites is an indication of importance, this is a very important problem: A Google search for 'Risk Return Portfolio Stock Exchange' produces about 10 million hits, among those http://www.ise.ie/intuition.asp?type=SUCCESS of the Irish stock exchange, where one can read that "In this section of the Exchange's e-learning tool you can learn more about the trade off between risk and return".

As the phrase 'trade off' indicates, portfolio optimisation is a classical bicriteria optimisation problem. It is arguably the first one that has been intensively studied since the seminal work of Markowitz [16] appeared. The original single objective

formulation employed for its solution is nothing but the ε-constraint scalarisation of the problem

$$\begin{aligned} \max f_1(x) &= \mu^T x \\ \min f_2(x) &= x^T \sigma x \\ \text{subject to } e^T x &= 1 \\ x &\geqq 0, \end{aligned} \tag{1}$$

where μ is the expected return, σ is the covariance matrix of the returns, and e is a vector of ones.

As a bi-objective programme with linear and quadratic objectives and linear constraints, the non-dominated and efficient sets are relatively easy to compute. Fig. 1 shows the non-dominated set of a portfolio optimisation problem with $n = 40$ assets from Ehrgott et al. [8].

So why should I talk about this problem? The reason becomes apparent when one compares theory with reality. As Konno [13] observes, most investors do not actually buy efficient portfolios, but rather those behind the non-dominated frontier. Can this behaviour be explained? The assumption underlying the Markowitz model is that investors are 'after the money' and therefore only interested in return and risk. One might call such investors 'average' or 'standard' investors. However, individual investors might not act according to the Markowitz assumption, and consider other, additional, objectives. Such *multi*objective models have been proposed, e.g. in [22] and [8]. The latter uses formulation (2)

$$\begin{aligned} \max f_1(x) &= \mu_1^T x \\ \min f_2(x) &= x^T \sigma x \\ \max f_3(x) &= \mu_3^T x \end{aligned}$$

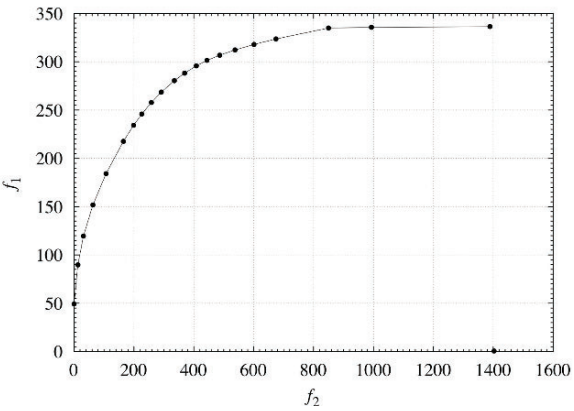


Fig. 1 The non-dominated frontier of a portfolio optimisation problem

$$\begin{aligned} \max f_4(x) &= d^T x \\ \max f_5(x) &= s^t x \\ \text{subject to } e^T x &= 1 \\ x &\geqq 0, \end{aligned} \tag{2}$$

where μ_1 and μ_3 are the 1- and 3-year expected returns, d is the dividend and s is the Standard and Poor star ranking. Steuer et al. [22] call investors that use such non-standard criteria 'suitable portfolio investors'. Investors may also like to have control over the number of assets in the portfolio and the fraction of investment in a single asset. This can be incorporated by using additional binary variables as in [3] to yield model (3).

$$\begin{aligned} \max f_1(x) &= \mu_1^T x \\ \min f_2(x) &= x^T \sigma x \\ \max f_3(x) &= \mu_3^T x \\ \max f_4(x) &= d^T x \\ \max f_5(x) &= s^t x \\ \text{subject to } e^T x &= 1 \\ x_i &\leqq u_i y_i, i = 1, \ldots, n \\ x_i &\geqq l_i y_i, i = 1, \ldots, n \\ e^T y &= k \\ y &\in \{0,1\}^n, \end{aligned} \tag{3}$$

where k is the number of assets, and l_i and u_i are lower and upper bounds on the fraction of capital invested in asset i.

By now, the continuous, convex, linear-quadratic bi-objective Markowitz model has become a true multiobjective and mixed integer problem, which is certainly worthy of further study. But besides showing that (multiobjective) portfolio optimisation remains an interesting topic more than 50 years after its first appearance one can learn another important lesson. Conventional portfolio theory cannot predict behaviour of individual investors. However, the introduction of additional objectives provides a rather plausible explanation of this phenomenon. The concept of suitable portfolio investors opens possibilities for further research. Assuming that investors make rational (optimal) decisions, how many and which objectives are needed to explain a particular solution as efficient? Furthermore, the importance of multicriteria decision aid increases, as criteria need to be made explicit and decision support is necessary to find an investor's most preferred portfolio.

Much more information on this topic can be found in Chap. 20 in [9] and references therein.

3 Transportation: Train Timetable Information

At the time of planning my trip to Europe, including attendance at the MOPGP conference, I considered using the train from Pirmasens, Germany, to Tours. The online timetable information of Deutsche Bahn (see http://www.reiseauskunft.bahn.de) provided two possible connections shown in Table 1.

So I had a choice between shorter travel time or fewer train changes. Obviously this is a multiobjective shortest path problem (a third objective, fare, is not available due to the international connection).

The multiobjective shortest path problem is a well studied multiobjective combinatorial optimisation (MOCO) problem. In particular, it is known that already the bi-objective version is NP-hard because the digraph at the top of Fig. 2 can be used to demonstrate a reduction from the NP-hard knapsack problem. Moreover, the graph at the bottom of Fig. 2 shows that it is intractable, i.e. there can be exponentially many efficient paths and non-dominated points, see, e.g. [5] for proofs.

This is pretty bad news for a problem which is so easy in its single objective version. Table 2 shows computation times on a standard PC and the number of efficient paths for relatively big networks from [19].

Apparently, the NP-hardness and intractability are not an issue in these examples (and all others tested in [19]). It is particularly striking that the large road networks have very few efficient paths. Can this discrepancy be explained?

Indeed it can. Mueller-Hannemann and Weihe [18] have investigated properties of networks with two objectives that allow better estimates of the number of efficient paths. Using the ratio between the first and second objective on the arcs they prove Theorem 1.

Table 1 Two train connections between Pirmasens and Tours

Station	Date	Time	Duration	Changes
Pirmasens	Su, 11.06.06	09:32	8:49	4
Tours	Su, 11.06.06	18:21		
Pirmasens	Su, 11.06.06	09:32	10:05	3
Tours	Su, 11.06.06	19:37		

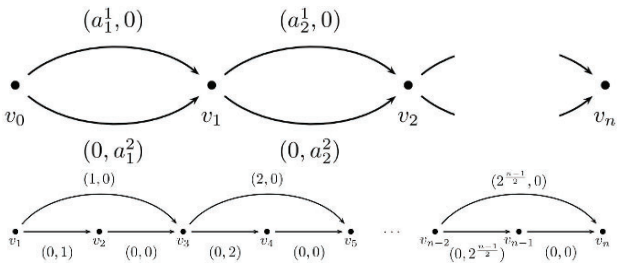


Fig. 2 The multiobjective shortest path problem is NP-hard (*top*) and intractable (*bottom*)

Table 2 Number of efficient paths and CPU times for bi-objective shortest path problems of different types

Type	Nodes	Edges	Efficient paths	CPU time
Grid	20,002	79,600	247	6.93
Grid	4,902	19,596	1,594	28.27
NetMaker	14,000	153,742	17	0.03
NetMaker	3,000	31,559	6	<0.01
Road	9,559	39,377	7	0.03
Road	330,386	1,202,458	21	1.10

Theorem 1.

(a) Even if the ratio between first and second length of an arc assumes only 2 values there can be exponentially many efficient paths.

(b) If there are k different ratios between first and second length of an arc there are at most $O(n^{2k-2})$ efficient bi-tonic paths. A bi-tonic path is a path where the sequence of ratios switches only once from increasing to decreasing.

Mueller-Hannemann and Weihe [18] have conducted experiments on the train graph of the Deutsche Bahn rail network, which has 1.4 million nodes, 2.3 million arcs and found that 84% of efficient paths are bi-tonic. Moreover, the number of efficient paths using different combinations of objectives is very small. For distance versus time on average two and at most eight paths are efficient. For fare versus time the numbers are three and 22 and for the three objectives distance, time, and train changes they are 10 and 96.

Again, one learns a number of lessons from this. Firstly, the concepts of NP-hardness may not be too relevant in multiobjective optimisation. Since almost all MOCO problems are NP-hard and intractable, there is virtually no distinction among problems by these criteria. Moreover, worst case estimates may simply not apply in a particular application, even if problem instances become very large. It is therefore always worthwhile studying the circumstances of the application. That will be beneficial for the application and it might lead to interesting mathematical results.

4 Transportation: Airline Crew Scheduling

BBC News of Sunday, 4 August, 2002, had an item that serves well to explain a problem in airline crew scheduling.

> Passengers with low-cost airline ... are suffering delays after 19 flights in and out of Britain were cancelled. The company blamed the move – which comes a week after passengers staged a protest sit-in at Nice airport – on crewing problems stemming from technical hitches with aircraft. Crews caught up in the delays worked up to their maximum hours and then had to be allowed home to rest. Mobilising replacement crews has been a problem as it takes time to bring people to airports from home. Standby crews were already being used and other staff are on holiday.

To understand how this problem arose, one needs to understand the standard integer programming model of the airline crew scheduling problem. The goal is to partition the scheduled flights into a set of 'pairings' each of which can be operated by a crew member to minimise cost. Let

$$a_{ij} = \begin{cases} 1 \text{ pairing } j \text{ includes flight } i \\ 0 \text{ otherwise.} \end{cases}$$

The problem can then be formulated as a generalised set partitioning problem

$$\begin{aligned} \min &= c^T x \\ \text{subject to } Ax &= e \\ Mx &= b \\ x &\in \{0,1\}^n. \end{aligned} \tag{4}$$

This particular type of set partitioning problem can be solved using column generation and constrained branching strategies [2]. Software to solve (4) (optimally or heuristically) is in use by all airlines. In fact, airline crew scheduling has been one of the biggest successes of Operations Research.

However, as can be seen from the news item above, things do not always go as planned and delays are common occurrences in operation. Optimal crew schedules according to (4) often operate with minimal 'sit times' between flights, that is without buffer time between flights to be operated by the same crew member. In addition, aircraft are also kept operating with minimal 'turn time' between flights. In such a scenario consider Fig. 3. If arriving flight F_r is late the next flight operated by the same aircraft is inevitably late, too. Moreover, if two crew members C and F arriving on flight F_r are scheduled to operate flights F_s and F_t, these will also be delayed. It is easy to imagine that this propagation of delays through the schedule can cause major and very expensive disruptions.

Thus, dealing with delay has become a focus of research in recent years. I shall explain two approaches. The first one is based on the stochastic nature of delays and incorporates the cost of delay into the problem formulation resulting in a stochastic programme with recourse [25]. The formulation is

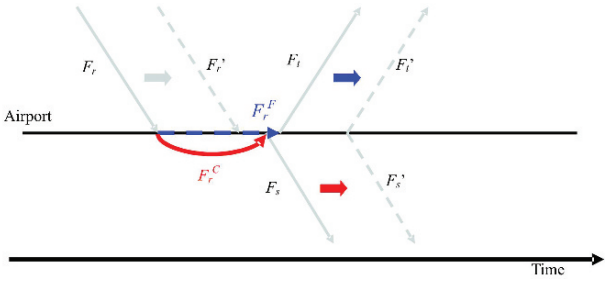


Fig. 3 Delay propagation due to aircraft changes

$$\begin{aligned} \min\, & c^T x + Q(x) \\ \text{subject to } & A_1 x = e \\ & A_2 x = b \\ & x \in \{0,1\}^n, \end{aligned} \tag{5}$$

where $Q(x) = \sum_{\omega \in \Omega} p(\omega) Q(x,\omega)$ and $Q(x,\omega)$ is the cost of delay under schedule x in scenario ω. Details of the solution algorithm – a branch and bound algorithm, which requires a set partitioning problem to be solved in every node – can be found in [25].

The second approach is based on the conflicting goals of minimising cost and minimising delay caused by aircraft changes, i.e. it is a bi-objective programme [6]:

$$\begin{aligned} \min\, & r^T x \\ \min\, & c^T x \\ \text{subject to } & A_1 x = e \\ & A_2 x = b \\ & x \in \{0,1\}^n, \end{aligned} \tag{6}$$

where r_j is a penalty for short ground time that does not allow recovery from previous delays. The 95%-quantile of the delay distribution is used as a delay measure in the calculation of r_j. The bi-objective set partitioning models are solved using the method of elastic constraints

$$\begin{aligned} \min\, & r^T x + ps \\ \text{subject to } & A_1 x = e \\ & A_2 x = b \\ & c^T x - s = \varepsilon \\ & x \in \{0,1\}^n \\ & s \geqq 0, \end{aligned} \tag{7}$$

a variant of the ε-constraint scalarisation which allows the ε-constraints to be violated but penalises the violation. It is known [7] that all solutions found are weakly efficient and that all efficient solutions can be found. But most importantly it turns out to be computationally superior to the ε-constraint method. In fact, an instance of (7) could be solved in approximately the same time as (4), whereas solving the ε-constraint scalarisation often exceeded the node limit of 1,000 and ran more than 10 times longer.

Both approaches have been implemented using the same crew scheduling software and the same schedule. The optimal solution of the stochastic programme and efficient solution of the bi-objective programme were simulated using 100 delay scenarios. Fig. 4 shows the average costs and delays.

Using either the stochastic or bi-objective approach to robust crew scheduling one may solve the problem of delays caused by aircraft changes. This does not

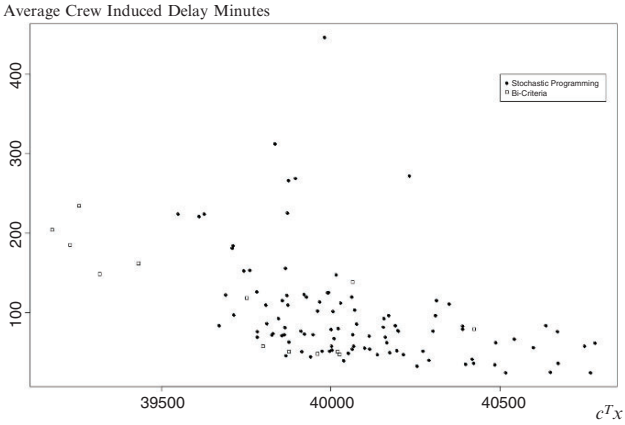


Fig. 4 Cost versus delay for schedules obtained with the stochastic and bi-objective programmes

address the problem of an arriving crew splitting up to operate different flights. Thus, the issue is only partially resolved: What is the use of having robust solutions for pilots if cabin crew do something different?

Robust crew scheduling should also address unit crewing, i.e. the problem of keeping crew together for a sequence of flights for as long as possible. Thus, one wants to solve the pairings problem for several crew groups simultaneously so as to minimise cost and maximise unit crewing. The corresponding problem formulation is

$$\begin{array}{lrrrrrl}
\min & c_1^T x_1 + & c_2^T x_2 & & & & \\
\min & & & e^T s_1 + & e^T s_2 & & \\
\text{subject to} & A_1 x_1 & & & & = & e \\
& M_1 x_1 & & & & = & b_1 \\
& & A_2 x_2 & & & = & e \\
& & M_2 x_2 & & & = & b_2 \\
& U_1 x_1 - & U_2 x_2 - & s_1 + & s_2 & = & 0 \\
& x_1 & x_2 & s_1 & s_2 & \in & \{0,1\}^n.
\end{array} \tag{8}$$

Using new branching strategies and the elastic constraint method as in (7), Tam [23] has obtained results that show that unit crewing, crew changing aircraft, and robustness of crew schedules are closely related, as shown in Fig. 5.

One sees here, that working on a particular application necessitates the development of new solution techniques. Such developments drive multiobjective programming, eventually making it applicable in other real word situations, where multiobjective models haven't been considered yet or couldn't be solved before. One can also see that bi-objective models may be an alternative to stochastic programming, if the recourse objective can be captured by means of a deterministic objective. This is especially relevant for integer models, where the stochastic programme is particularly hard to solve and computationally expensive. Finally, multiobjective modelling

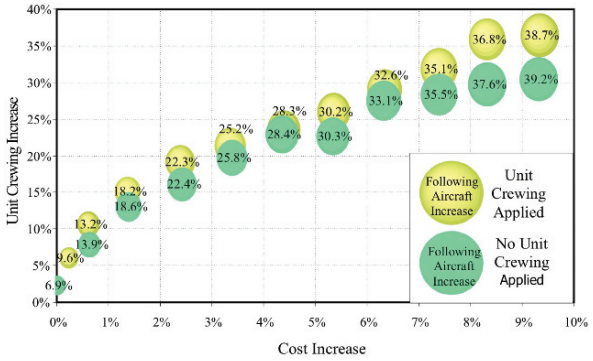


Fig. 5 Unit crewing versus number of aircraft changes and cost

can lead to a more comprehensive view of an application. From the bi-objective model for robust scheduling it is only a small step towards the simultaneous consideration of several crew groups in the unit crewing problem. In fact, the next question is immediate: Why not include aircraft routing and consider assignment of aircraft to flights simultaneously with crew scheduling? This is an unsolved problem that is currently under investigation [24].

5 Medicine: Radiotherapy Treatment Design

External beam radiotherapy is one of the major forms of cancer treatment. About 50% of cancer patients receive radiotherapy for curative or palliative purposes. Beams of electrons or high energy photons generated by a linear accelerator are focused on the tumour from several directions. An oncologist prescribes a dose distribution to be achieved by the treatment, that is a radiation dose to be delivered to the tumour that achieves the curative or palliative intent of the treatment but avoids damage to healthy tissues.

Given the beam directions, the purpose of radiotherapy treatment design is to find beam intensity (or fluence) maps for each beam that realise the desired dose distribution. Here I consider the treatment design problem for intensity modulated radiotherapy (IMRT), where beam intensity can vary across a beam. The advantage of IMRT is described on http://www.cancernews.com/data/Article/259.asp as follows

> IMRT represents an advance in the means that radiation is delivered to the target, and it is believed that IMRT offers an improvement over conventional and conformal radiation in its ability to provide higher dose irradiation of tumour mass, while exposing the surrounding normal tissue to less radiation.

Many optimisation models, both linear and nonlinear, are available for this problem. The most popular optimisation model is based on an oncologist's prescription

of a goal dose TG to the target and upper bounds CG and NG on the dose to critical structures and normal tissue. It consists of the minimisation of a norm of the (nonnegative) deviation of delivered and goal dose:

$$\min_{x \geqq 0} \omega_T \|A_T x - TG\| + \omega_C \left\|(A_C x - CG)_+\right\| + \omega_N \left\|(A_N x - NG)_+\right\|, \tag{9}$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. A_T, A_C and A_N are dose deposition matrices which allow to calculate the dose distribution generated by intensity x. In practice the Euclidean norm is most often used and the most popular solution technique is simulated annealing. The result of the optimisation depends crucially on the values of $\omega_T, \omega_C, \omega_N$, which are often deemed indispensable for effective treatment planning. A trial and error process is usually needed to find values that result in a good quality treatment.

The words 'higher' and 'lower' in the above quotation indicate that treatment planning is about conflicting goals. And to anyone familiar with multiobjective optimisation it is obvious that the standard dose based model (9) is the weighted sum scalarisation of the multiobjective programme

$$\min_{x \geqq 0} \left(\|A_T x - TG\|, \left\|(A_C x - CG)_+\right\|, \left\|(A_N x - NG)_+\right\| \right). \tag{10}$$

However, this model has only been used in the form (9), with a set of pre-selected weights to produce several efficient plans [4, 15]. The first non-scalarised multiobjective LP model has been proposed by Küfer and Hamacher [14].

In the context of the multitude of objective functions used in radiotherapy treatment planning models a theorem stated in [20] becomes important.

Theorem 2. *The two multiobjective problems* $\min\{(f_1(x), \ldots, f_p(x)) : x \in X\}$ *and* $\min\{(h_1(f_1(x)), \ldots, h_p(f_p(x))) : x \in X\}$ *with strictly increasing functions* $h_1, \ldots, h_p$ *are equivalent.*

Theorem 2 is not really surprising, but it is important as it illustrates that much of the discussion about the right model is void. I present here a linear model with three objectives derived from (the scalar) LP in [12]:

$$\begin{aligned} \min(z_T, z_C, z_N) & \\ \text{subject to } A_T x + z_T e &\geqq l_T \\ A_T x &\leqq u_T \\ A_C x - z_C e &\leqq u_C \\ A_N x - z_N e &\leqq u_N \\ z_N &\geqq 0 \\ x &\geqq 0. \end{aligned} \tag{11}$$

This multiobjective linear programme may have thousands of variables and tens of thousands of constraints. Since it has only three objectives it is advantageous

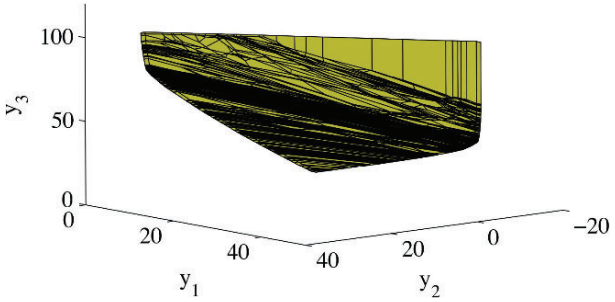


Fig. 6 The non-dominated set of (11) for a prostate cancer example

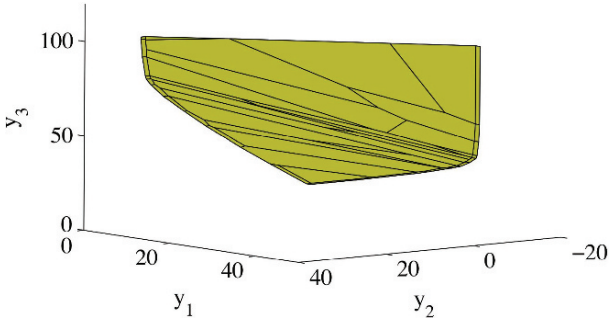


Fig. 7 The approximate non-dominated set of (11) for the same prostate cancer example

to solve it in objective space. Benson's "outer approximation" algorithm [1] can be used for this purpose. In [21] a simplified version of this problem with 1,293 constraints and 821 variables has been solved. 3,165 non-dominated extreme points have been obtained, shown in Fig. 6. The computation took nearly 1 h.

From Fig. 6 it can already be seen that many of the extreme points differ only very slightly. This result points to another issue: How is a solution to be selected among so many options? And does one want an extreme point solution in the first place?

At this stage it is necessary to reconsider the model. It uses a dose deposition matrix A, separated by rows into A_T, A_C, and A_N as input. The entries in A are the result of a dose calculation model, that calculates the amount of dose deposited at a point in the body at unit intensity of a sub-beam. Even with sophisticated dose models this calculation is imprecise. That means that the data of (11) is imprecise. It turns out that in clinical practice calculating a dose distribution to 0.1 Gy precision is sufficient.

It is therefore possible to use an approximation version of Benson's algorithm that is guaranteed to solve the MOLP (11) to within an additive precision of 0.1 and achieve a dramatic effect. This algorithm calculates 88 non-dominated extreme points in less than one minute. The approximated non-dominated set is shown in Fig. 7.

I have so far assumed that the beam directions are given. However, they also have to be chosen. This choice is currently done manually. Mathematically the optimisation of beam directions can easily be incorporated in (11):

$$\begin{aligned} &\min(z_T, z_C, z_N) \\ \text{subject to } &A_T x + z_T e \geqq l_T \\ &A_T x \leqq u_T \\ &A_C x - z_C e \leqq u_C \\ &A_N x - z_N e \leqq u_N \\ &z_N \geqq 0 \\ &x \geqq 0 \\ &x \leqq My \\ &e^T y \leqq r. \end{aligned} \tag{12}$$

The solution of the large scale multiobjective mixed integer programme (12) is a challenge for both multiobjective optimisation and radiotherapy treatment design.

I find this application particularly instructive. It shows how hard it can be to convince practitioners of the usefulness of multiobjective optimisation, even if they already use elements of it, albeit unknowingly. It is a reminder that the results of optimisation can never be more precise that the input data, and that it is always worth exploiting features of the application to simplify methods. In addition, applying multiobjective optimisation can lead to improved processes in the application area as secondary benefits. In this example the trial and error search for 'optimal' weights can be eliminated. Instead, treatment planners can concentrate on their main task, namely to find a best possible treatment plan for the patient. Because the multiobjective model allows a separation of plan calculation and selection, a speed up of the planning process can be expected. Again, multicriteria decision aid is called upon to provide appropriate decision support systems.

6 Telecommunication: Routing in IP Networks

Routing of data packets in computer networks using the internet protocol is usually based on the OSPF protocol (open shortest path first). This protocol applies Dijkstra's algorithm to minimise the number of hops (the number of intermediate routers) along the path from the origin of the packet to its destination. While other protocols exist that allow aggregation of several objectives, routing is still using a 'best effort' rather than 'Quality of Service' philosophy.

It does not take much imagination to see that several objectives are relevant in this context. Gandibleux et al. [10] have developed a routing protocol that uses the three objectives

- $\min f_1(p) = \sum_{(i,j)\in p} c^1(i,j)$, where $c^1(i,j)$ denotes the delay on link (i,j)
- $\max f_2(p) = \min_{(i,j)\in p} c^2(i,j)$, $c^2(i,j)$ denoting the available bandwidth of link (i,j) and
- $\min f_3(p) = |\{(i,j) \in p\}|$, counting the number of hops

as well as additional constraints. They have implemented a modification of Martin's label correcting algorithm [17] to deal with the constraints and the bandwidth objective, which is of the min max rather than the min sum type.

Considering the delay and bandwidth objectives only, there are five efficient paths from node seven to node eleven in the network of Fig. 8, an actual IP network (bandwidth and delay are listed along the arcs).

This application shows that even long known algorithms can be useful in today's problems. Once more, as seen in the other examples, multiobjective modelling helps thinking outside the box.

Chapter 22 in [9] and references therein contain much more on multicriteria decision analysis in telecommunication.

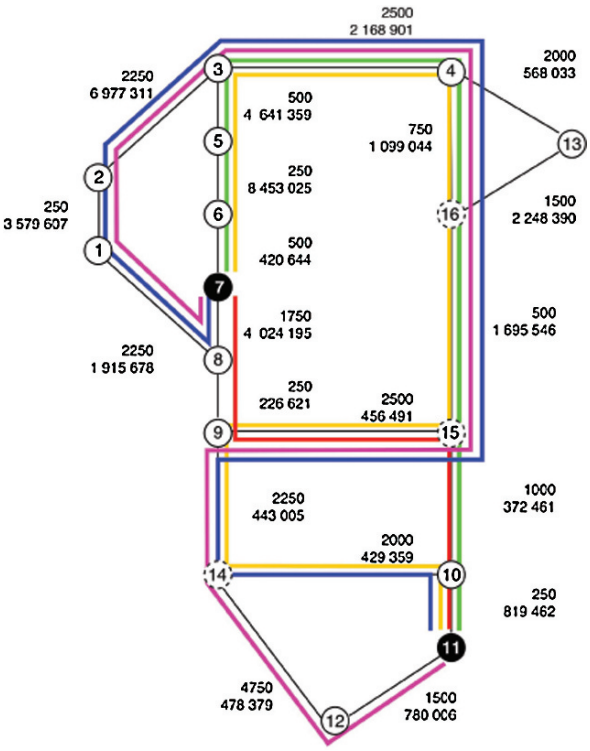


Fig. 8 Efficient routes in an IP network using bandwidth and delay objectives

7 Conclusion

In this paper I have sketched a number of applications of multiobjective programming. I have tried to show that interacting with practitioners in many areas is mutually beneficial in the sense that real world applications provide opportunities for progress in multiobjective optimisation methodology and theory and that multiobjective models provide insights in applications that conventional models cannot reveal. In particular, multiobjective models help question standard procedures and thus induce the practitioner to think outside a conventional framework. It is easily possible that multiobjective optimisation results in benefits that are not at all part of the model. Last but not least, the real world has many challenges and new application areas in store to motivate established and future researchers to work in this area.

References

1. Benson H (1998) An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. J Global Optim 13(1):1–24
2. Butchers E, Day P, Goldie A, Miller S, Meyer J, Ryan D, Scott A, Wallace C (2001) Optimised crew scheduling at Air New Zealand. Interfaces 31(1):30–56
3. Chang T, Meade N, Beasley J, Sharaiha Y (2000) Heuristics for cardinality constrained portfolio optimisation. Comput Oper Res 27:1271–1302
4. Cotrutz C, Lahanas M, Kappas C, Baltas D (2001) A multiobjective gradient-based dose optimization algorithm for external beam conformal radiotherapy. Phys Med Biol 46:2161–2175
5. Ehrgott M (2005) Multicriteria optimization, 2nd edn. Springer, Berlin
6. Ehrgott M, Ryan D (2002) Constructing robust crew schedules with bicriteria optimization. J Multicrit Decision Anal 11:139–150
7. Ehrgott M, Ryan D (2003) The method of elastic constraints for multiobjective combinatorial optimization and its application in airline crew scheduling. In: Tanino T, Tanaka T, Inuiguchi M (eds) Multi-objective programming and goal-programming. Springer, Berlin, pp 117–122
8. Ehrgott M, Klamroth K, Schwehm C (2004) An MCDM approach to portfolio optimization. Eur J Oper Res 155:752–770
9. Figueira J, Greco S, Ehrgott M (eds) (2005) Multiple Criteria Decision Analysis: State of the Art Surveys, vol 78 of Springer's International Series in Operations Research and Management Science. Springer Science and Business Media, New York
10. Gandibleux X, Beugnies F, Randriamasy S (2006) Martins' algorithm revisited for multiobjective shortest path problems with a maxmin cost function. 4OR 4:47–59
11. Hansen P (1979) Bicriterion path problems. In: Fandel G, Gal T (eds) Multiple Criteria decision making theory and application, vol 177 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, pp 109–127
12. Holder A (2003) Designing radiotherapy plans with elastic constraints and interior point methods. Health Care Manage Sci 6:5–1
13. Konno H (1990) Piecewise linear risk function and portfolio optimization. J Oper Res Soc Jpn 33:139–156
14. Küfer K-H, Hamacher H (2000) A multicriteria optimization approach for inverse radiotherapy planning. In: Schlegel W, Bortfeld T (eds) XIIIth International Conference on the Use of Computers in Radiation Therapy. Springer, Berlin, pp 26–28

15. Lahanas M, Schreibmann E, Baltas D (2003) Multiobjective inverse planning for intensity modulated radiotherapy with constraint-free gradient-based optimization algorithms. Phys Med Biol 48:2843–2871
16. Markowitz H (1952) Portfolio selection. J Finance 7:77–91
17. Martins E (1984) On a multicriteria shortest path problem. Eur J Oper Res 16:236–245
18. Müller-Hannemann M, Weihe K (2006) On the cardinality of the Pareto set in bicriteria shortest path problems. Ann Oper Res 147:269–286
19. Raith A, Ehrgott M (2008) A comparison of solution strategies for biobjective shortest path problems. Computers & Operations Research doi 10.1016/j.cor.2008.02.002
20. Romeijn H, Dempsey J, Li J (2004) A unifying framework for multi-criteria fluence map optimization models. Phys Med Biol 49:1991–2013
21. Shao L, Ehrgott M (2008) Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. Mathematical Methods of Operations Research 68:257–276
22. Steuer R, Qi Y, Hirschberger M (2007) Suitable-portfolio investors, nondominated frontier sensitivity, the effect of multiple objectives on standard portfolio selection. Ann Oper Res 152:297–317
23. Tam B (2004) Multi-criteria methods for unit crewing in the airline tour-of-duty planning problem. In: Ehrgott M (ed) Proceedings of the 39th Annual Conference of the Operational Research Society of New Zealand. pp 174–183 Available online at http://www.orsnz.org.nz/conf39/Papers/TamB.pdf.
24. Weide O, Ryan D, Ehrgott M (2007) Iterative airline scheduling. Technical Report 645, Department of Engineering Science, The University of Auckland, New Zealand
25. Yen J, Birge J (2006) A stochastic programming approach to the airline crew scheduling problem. Transport Sci 40(1):3–14

Multi-scenario Multi-objective Optimization with Applications in Engineering Design

Margaret M. Wiecek, Vincent Y. Blouin, Georges M. Fadel, Alexander Engau, Brian J. Hunt, and Vijay Singh

Abstract The notion of multi-scenario multi-objective optimization is proposed as a methodological framework for handling engineering design and other decision problems represented as a collection of multi-criteria optimization problems. Three specific research issues are discussed in this context, namely, the modelling of decision maker's preferences, the development of a concept of optimality, and the development of solution approaches to finding a preferred feasible solution for the overall problem. Two models of preferences that generalize the classical Pareto preference and two solution approaches to a class of multi-scenario multi-objective optimization problems are presented. Illustrative examples are included.

Keywords: Engineering design · Multi-objective optimization · Multi-scenario optimization · Preferences

1 Introduction

The science of engineering design assumes a decision-making paradigm for the product development process, which is substantiated by the application of mathematical optimization as a tool for modelling and solving the underlying decision problems. Because of the multiplicity of engineering disciplines and criteria involved in a design problem, design has been studied within the framework of multidisciplinary optimization and multi-objective optimization.

Mass customization has forced the engineering and science communities to look for designs appropriate for groups of applications, rather than a single application. For example, while customers typically want a car with maximum reliability at

M.M. Wiecek (✉)
Departments of Mathematical Sciences and Mechanical Engineering, Clemson University, Clemson, SC 29634, USA
e-mail: wmalgor@clemson.edu

V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, © Springer-Verlag Berlin Heidelberg 2009

minimum cost, a higher level of reliability usually results in a higher cost. This design problem can be modelled as an optimization problem with two conflicting criteria. Additionally, the car may be designed for various driving conditions, different markets, or different types of use. While in every scenario the criteria may have different mathematical representations, their physical interpretation remains the same, that of minimizing cost and maximizing reliability. In other applications, the mathematical representation, the physical meaning of the criteria, and the design space may vary from scenario to scenario. As a result, a design process under different scenarios for the same physical problem leads to the problem of *multi-scenario multi-objective* optimization.

The concept of *scenario* has not been formally defined in the mathematical literature, although the context of some research efforts brings an obvious analogy to this concept. For example, in their study of single-objective optimization problems, Kouvelis and Yu [13] propose the concept of "robust solution," one that would be robust for the same mathematical model associated with multiple data instances or scenarios. As a result, finding the robust design is based on multi-objective optimization.

In the field of engineering, load cases are commonly considered in structural analysis. A structure to be designed will be subjected to different loads or forces, and the analysis proceeds by examining the structural response for each load case. The objective is typically to minimize weight, and the stress constrains the solution space. The worst case scenario is used to dimension the structural members. In such an example, the load cases are scenarios.

Problems in multi-disciplinary optimization mentioned above may also lead to multi-scenario multi-objective optimization if a multi-objective problem is associated with every design discipline. For example, this is the case in aircraft wing design in which three engineering design disciplines are involved (aerodynamics, structural mechanics, and control), and a scenario is understood as a design discipline [18].

Another application offering the context of scenarios comes from product family design [7]. Traditional design process considers designing a single product, while product family design, studied since the 1990s, deals with groups of related products. A product platform is a set of common components or parts from which several variations of a product can be made. Product platform design requires selection of shared parts and assessment of potential sacrifices in individual product performance resulting from parts sharing. Each product can be viewed as a scenario. Depending on the number of products in a family, a single or multi-scenario problem can be formulated. For example, if there are two products in the family, a bi-scenario problem can be formulated by associating some performance criteria with each of the products.

In vehicle design, recent efforts have focused on the optimization of groups of vehicle performance indices (criteria) in different operating manoeuvres (scenarios) [6, 9].

This paper reports on a multi-year joint research effort conducted at Clemson University between a group of operations researchers and a group of engineering

design researchers [6, 17]. The latter raised the need for the consideration of multi-scenario multi-objective optimization, provided a host of relevant engineering problems and case studies that revealed new research issues, while the former undertook an effort to develop a supporting theory. In this paper we propose a methodological framework for handling engineering design and other problems represented as a collection of multi-criteria problems and thus formalize the concept of multi-scenario multi-objective optimization.

While the concept of optimality for conventional multi-criteria problems is well established and researched in the literature, the same concept for a collection of such problems remains unknown. In Sect. 2, we define a multi-scenario multi-objective program. We then focus on two specific research issues arising in this context, namely, modelling decision makers (DMs) preferences over a set of scenarios and the development of solution approaches to finding a preferred feasible solution for the overall problem. In Sect. 3, we present two models of preferences that generalize the classical Pareto preference and apply one of them to a tractor–trailer design problem. In Sect. 4, we propose two solution approaches to a class of multi-scenario multi-objective optimization problems and illustrate one of them on a mathematical example. The paper is concluded in Sect. 5.

2 Multi-Scenario Multi-Objective Optimization

The multi-objective programming framework includes the following basic elements: a set of feasible solutions (decisions), a collection of objective (criterion) functions (performance indices) that evaluate the solutions and produce attainable outcomes, and DMs preferences. The goal is to identify those feasible solutions that yield the most satisfactory (preferred) outcome(s) according to the DM's preferences. In our work we use the concept of optimality introduced by Yu [20] who extends the classical definition of Pareto-optimality [15] and uses convex cones to model DMs preferences.

Let $\mathbb{R}^n$ be an n-dimensional Euclidean decision space and n the number of decision variables. Let $S = \{1, \ldots, N\}$ be a set of scenarios, $X^s \subseteq \mathbb{R}^n$ be the set of feasible solutions for scenario $s, s \in S$, with $X = \cap_s X^s \neq \emptyset$, and C^s be a cone modelling DMs preferences for scenario s. The multi-scenario multi-objective program (MSMOP) is represented by the following collection of multi-objective programs:

$$\{(X^s, f^s, C^s), s \in S\}, \tag{1}$$

where $f^s = [f_{s1}, f_{s2}, , f_{sm(s)}]$, and $f_{sj} : \mathbb{R}^n \to \mathbb{R}, j = 1, 2, \ldots, m(s)$ are objective (criterion) functions associated with scenario s, and $m(s)$ is the number of objective functions in scenario s.

Consider a single-scenario multi-objective program $\text{MOP}^s = (X^s, f^s = [f_{s1}, f_{s2}, \ldots, f_{sm(s)}], C^s)$, or for brevity given as $\text{MOP} = (X, f = [f_1, f_2, \ldots, f_m], C)$. Let $\mathbb{R}^m$ be referred to as the objective space and define the set of outcomes $Y \subseteq \mathbb{R}^m$ as

the set $Y = \{y \in \mathbb{R}^m : y = f(x) \text{ for } x \in X\}$. Let $C \subseteq \mathbb{R}^m$ be a convex polyhedral cone. Assume that C is a set of all dominated directions in $\mathbb{R}^m$ and refer to it as the domination cone. A domination cone contains all vectors $d \in \mathbb{R}^m$ such that for $x, x^1 \in X$, if $f(x^1) = f(x) + d$ for some $d \in D, d \neq 0$, then $f(x^1)$ is dominated by $f(x)$. The vectors in the domination cone can be thought of as "bad" or "dominated" directions to travel within $\mathbb{R}^m$. To solve MOP is understood as to find its *efficient set* $E(X, f, C)$ in X. A feasible solution $x \in E(X, f, C)$ if there does not exist another feasible solution $x^1 \in X$ and $d \in C, d \neq 0$, such that $f(x) = f(x^1) + d$. The image of $E(X, f, C)$ is referred to as the *non-dominated set* $N(Y, C)$ and a non-dominated outcome (element of $N(Y, C)$) is one that is not dominated by any other outcome in Y. We also make use of the *weakly efficient set* denoted as $w\text{-}E(X, f, C)$ and the ε*-efficient set* denoted $\varepsilon\text{-}E(X, f, C)$. Let int$C$ denote the interior of the cone C. A feasible solution $x \in w\text{-}E(X, f, C)$ if there does not exist another feasible solution $x^1 \in X$ and $d \in \text{int}C$, such that $f(x) = f(x^1) + d$. Let $\varepsilon \in C$. A feasible solution $x \in \varepsilon\text{-}E(X, f, C)$ if there does not exist another feasible solution $x^1 \in X$ and $d \in C$, such that $f(x) - \varepsilon = f(x^1) + d$. The *weakly non-dominated set*, denoted $w\text{-}N(Y, C)$, and the ε*-non-dominated set*, denoted $\varepsilon\text{-}N(Y, C)$, are defined accordingly.

For the concept of Pareto-optimality, that is commonly used in multi-objective optimization, the polyhedral Pareto (domination) cone, C^{Par}, is given as $C^{\text{Par}} = \{d \in \mathbb{R}^m : d \geq 0\}$. In this case, the sets $E(X, f, C^{\text{Par}})$ and $N(Y, C^{\text{Par}})$ are referred to as Pareto efficient and Pareto non-dominated, respectively. Given two outcomes $y^1, y^2 \in Y, y^1 \neq y^2$, with $y^1 \geq y^2$, we say that y^1 is *Pareto dominated* by y^2 or that y^2 is *Pareto dominating* y^1.

The Pareto preference can be generalized by applying a $p \times m$ matrix A in the cone description, which gives a new polyhedral cone

$$C^A = \{d \in \mathbb{R}^m : Ad \geq 0\}, \tag{2}$$

with the matrix A becoming the algebraic model of the new preference. While the terminology efficient and non-dominated is used for preferences with general cones, the terminology A-efficient and A-non-dominated is being used for preferences modelled with polyhedral cones represented by the matrix A as given in (2).

For a review of recent advances in cone-based preference modelling for decision making with multiple criteria the reader is referred to [19].

3 Modelling Preferences

A review of multi-objective optimization literature indicates that weighting methods [8] and ranking methods (e.g., based on the lexicographic order) are probably most commonly used to model relative importance of criteria in real-life applications. A different approach based on cones is proposed by Noghin [14] who uses weights to augment the Pareto cone and model importance of criteria. Hunt and Wiecek [11] follow on Noghin and advocate the use of cones in a simple design problem. In

the same spirit, Hunt [10] develops a theoretical and methodological framework to model specific cone-based preferences. Since in general, a different preference may guide each MOP in the multi-scenario formulation, we employ this framework to allow for that modelling feature.

3.1 Algebraic Models of Preferences

We now consider a simplified formulation of MSMOP in which in every scenario the criterion functions are the same but cones modelling DMs preferences may be different. In other words, while the same criteria are used across all scenarios for the evaluation of feasible decisions, the distinction among the criteria is captured in their relative importance that might be different from scenario to scenario.

Consider again MOP = $(X, f = [f_1, f_2, \ldots, f_m], C)$ and assume that the DM initially chooses C to represent the Pareto preference according to which all objective functions are to be minimized. However, the DM additionally considers certain criteria more important than the others and, due to that importance, is willing to accept decay in the latter to achieve improvement in the former. We propose two approaches to model this relative importance by means of convex polyhedral cones that subsume the Pareto cone.

In Model 1, only one criterion i, $i \in \{1, ..., m\}$ is assumed to be less important than the other criteria and therefore it is allowed to decay (increase its values) while all the other criteria, being more important, improve (decrease their values). We define allowable trade-offs $a_{ij} \geq 0$ for every $j \in \{1, \ldots, m\}, j \neq i$, representing the number of units of decay in criterion i for 1 unit of improvement in criterion j. The $m(m-1) \times m$ matrix A modelling this preference consists of m blocks, each with $m-1$ rows and m columns, because for each criterion viewed as less important, the remaining $m-1$ criteria improve. Let A^{ij} denote row $j \in \{1, \ldots, m-1\}$ of block $i \in \{1, \ldots, m\}$, and $(A^{ij})_k$ denote the element of A^{ij} in column $k \in \{1, \ldots, m\}$. Then $(A^{ij})_i = 1$ for all i and j, and $(A^{ij})_j = a_{ij}$ if $j < i$. Also, $(A^{ij})_{j+1} = a_{i(j+1)}$ if $j > i$, and $(A^{ij})_k = 0$ otherwise. Figure 1 depicts the matrices of Model 1 for multi-objective programs with 2, 3, and 4 criteria.

In the more general Model 2, all criteria are divided into two groups: the group of more important criteria, M, and the group of less important criteria, L. We assume that the maximum decay allowed for each criterion in L is bounded by the total improvement for all criteria in M. We define allowable trade-offs a_{ij} for every pair $(i, j), i \neq j, i \in L, j \in M$, where $a_{ij} \geq 0$ denotes the number of units of decay in criterion i for one unit of improvement in criterion j. The structure of the $m \times m$ matrix modelling this preference depends on the sets L and M. Figure 2 illustrates these matrices for multi-objective problems with two and three criteria. For the derivation of both algebraic models and the properties of the cones, the reader is referred to [10].

$$\begin{bmatrix} 1 & a_{12} \\ a_{21} & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & a_{12} & 0 \\ 1 & 0 & a_{13} \\ a_{21} & 1 & 0 \\ 0 & 1 & a_{23} \\ a_{31} & 0 & 1 \\ 0 & a_{32} & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & a_{12} & 0 & 0 \\ 1 & 0 & a_{13} & 0 \\ 1 & 0 & 0 & a_{14} \\ a_{21} & 1 & 0 & 0 \\ 0 & 1 & a_{23} & 0 \\ 0 & 1 & 0 & a_{24} \\ a_{31} & 0 & 1 & 0 \\ 0 & a_{32} & 1 & 0 \\ 0 & 0 & 1 & a_{34} \\ a_{41} & 0 & 0 & 1 \\ 0 & a_{42} & 0 & 1 \\ 0 & 0 & a_{43} & 1 \end{bmatrix}$$

Fig. 1 Matrices of Model 1 for $m = 2$, 3, and 4

$$\begin{bmatrix} 1 & 0 \\ a_{21} & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & a_{12} \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & a_{12} & 0 \\ 0 & 1 & 0 \\ 0 & a_{32} & 1 \end{bmatrix}$$

(a) (b) (c) (d)

Fig. 2 Matrices of Model 2: (**a**) $m = 2, M = \{1\}, L = \{2\}$; (**b**) $m = 2, M = \{2\}, L = \{1\}$; (**c**) $m = 3, M = \{1,2\}, L = \{3\}$; (**d**) $m = 3, M = \{2\}, L = \{1,3\}$

3.2 Generating A-Efficient Solutions

Two results from the literature help establish methods for finding A-efficient solutions $E(X, f, C^A)$ and A-non-dominated outcomes $N(Y, C^A)$.

Theorem 1 ([16]) *Let C_1 and C_2 be cones in $\mathbb{R}^m$. If $C_1 \subseteq C_2$ then $N(Y, C_2) \subseteq N(Y, C_1)$.*

Theorem 2. *Let C be a convex and pointed cone in $\mathbb{R}^m$ represented by $C^A = \{d \in \mathbb{R}^m : Ad \geq 0\}$. Then*

(a) [14] $E(X, f, C^A) = E(X, Af, C^{Par})$;
(b) [11] $A[N(Y, C^A)] = N(A[Y], C^{Par})$.

Based on Theorem 2, given two outcomes $y^1, y^2 \in Y, y^1 \neq y^2$, with $Ay^1 \geq Ay^2$, we say that y^1 is *A-dominated* by y^2 or that y^2 is *A-dominating* y^1.

Two methods are proposed to implement the preference models and generate A-efficient solutions. In the first one-step method, given a matrix A modelling the preferences we find the set $E(X, Af, C^{\text{Par}})$. In the other two-step method, we first find the set $E(X, f, C^{\text{Par}})$ and then the set $E(E(X, f, C^{\text{Par}}), Af, C^{\text{Par}})$. Each method requires finding the Pareto efficient set with respect to new criterion functions that result from applying the matrix A to the original criteria. Clearly, to apply the models, the DM is expected to have knowledge and experience to come up with the

allowable trade-offs a_{ij} and construct the matrices. Otherwise, the DM may interactively apply the models with different trade-offs and learn how they affect the resulting A-efficient set.

3.3 Application to Tractor–Trailer Design

Preference Model 2 is now applied to the design of a tractor–trailer vehicle for optimum dynamic behaviour. This engineering problem has been selected to demonstrate a standard difficulty in engineering optimization. The design problem is formulated as an optimization problem that, due to modelling limitations, can only be solved with numerical simulation rather than conventional mathematical optimization. Additionally, we assume that the DM is not willing to come up with specific allowable trade-offs and will engage in the interactive learning process.

The tractor–trailer is a six-axle articulated heavy vehicle and is optimized for a standard manoeuvre, namely, the single lane change manoeuvre [2]. Twenty-one design variables representing physical parameters of the vehicle include, among others, the tire stiffness, the locations of the centres of gravity (COG) of the tractor and the trailer, the wheel-base length, and the track widths as illustrated in Fig. 3.

We consider three criteria (or performance indices) to be minimized, namely, the rearward amplification factor (RWA), the load transfer ratio (LTR), and the high-speed friction demand (FD). RWA is the ratio of the peak lateral acceleration of the mass centre of the trailer to that developed at the mass centre of the tractor during the manoeuvre. LTR corresponds to the ratio of the absolute value of the difference between the sum of the right wheel loads and the sum of the left wheel loads to the sum of all the wheel loads. Finally, FD measures the force utilized by the tractor drive axle to overcome the trailer aligning moment for the total drive axle tire adhesion force during the manoeuvre without reaching the full skid condition of the drive axle. These three criteria are generally in conflict, making the design problem challenging to designers. The numerical model used in this research is the ArcSim tractor–trailer model developed at the University of Michigan [1].

In place of mathematical optimization, a Latin hypercube sampling is used to solve the resulting multi-criteria optimization problem. The sampling of the feasible space produces 1,000 well distributed points located within $+/-$ 40% of a baseline design. For each point, a time-dependent numerical simulation is performed, upon

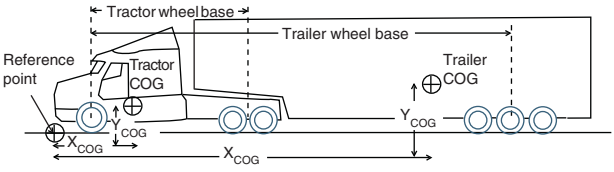


Fig. 3 Illustration of a tractor–trailer and design parameters

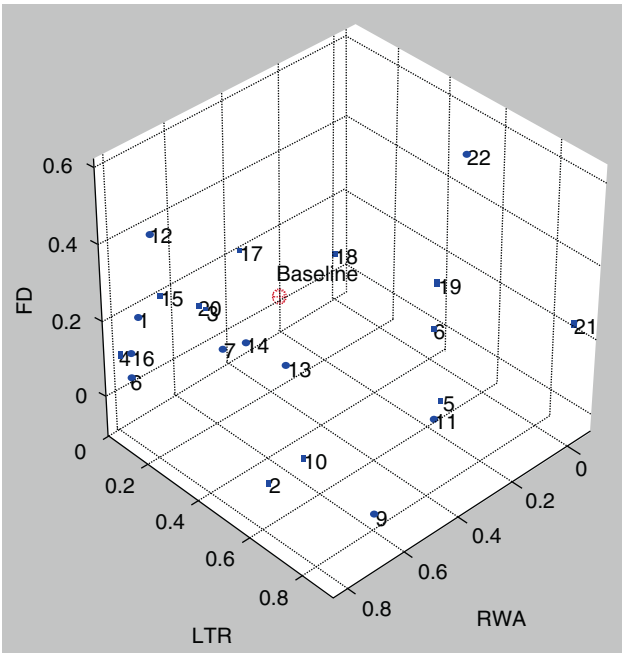


Fig. 4 Pareto non-dominated outcomes for the tractor–trailer design problem

$$\begin{bmatrix} 1 & a_{12} & a_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ a_{21} & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}$$

Fig. 5 Matrices $A_t, t = 1$ (*left*), 2 (*centre*), 3 (*right*) for Model 2

which the criterion functions are computed and then normalized to avoid numerical difficulties caused by differences in scale. The two-step method is used to find A-non-dominated outcomes. Within the 1,000 points, 22 outcomes are determined to be Pareto non-dominated and used as an approximation of the actual Pareto-non-dominated set. Figure 4 depicts these outcomes in the normalized objective space.

The preferences are defined by three matrices $A_t, t = 1,2,3$, each constructed by a pair of allowable tradeoffs $(a_{12}, a_{13}), (a_{21}, a_{23}), (a_{31}, a_{32})$, respectively, as shown in Fig. 5.

Figure 6 depicts two views of three piecewise linear surfaces representing the number of A-non-dominated outcomes after the corresponding preference has been applied. Each of the surfaces is generated when the allowable tradeoffs in one of the three matrices are independently varied between 0 and 1. Figure 6 also shows the impact of each preference (with normalized values a_{ij}) on the reduction of the set of Pareto non-dominated outcomes. In this example, matrix A_1 leads to the largest reduction, while matrix A_3 leads to the smallest reduction.

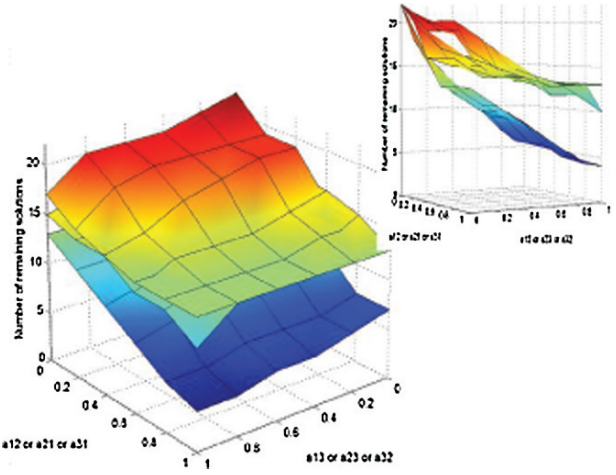


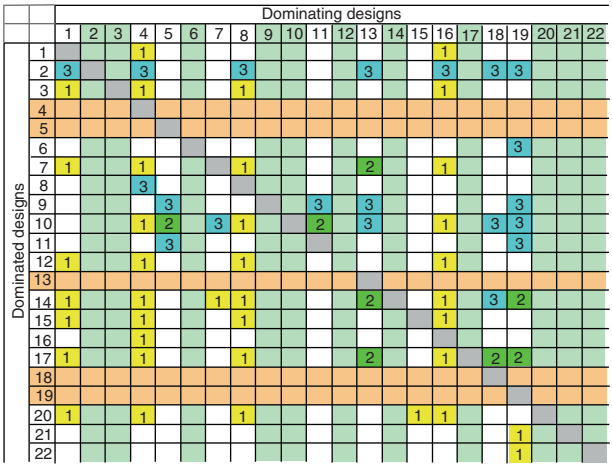
Fig. 6 Number of A_t-non-dominated outcomes after applying Model 2 with various allowable trade-off values to 2the trailer design problem

Table 1 Dominating and dominated outcomes for tractor–trailer design ($a_{ij} = 1.0$)

Dominated designs \ Dominating designs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1				1												1						
2	3		3	3			3	3	2			3	3	3	3	3	3	3	3	3		3
3	1			1	2	2	2	1			2		2			1			2		2	
4																						
5				1				1											3			
6	1			1				1					1			1			3			
7	1			1	2			1			2		2			1						
8				3												3						
9					3	3					3		3				3	3	3			3
10		1	3	1	2		3	1	2		2		3	3		1	3	3	3	3		3
11					3	3		1										3	3			
12	1		2	1	2	2	2	1			2		2			1		2	2		2	
13				1	2			1								1		3				
14	1			1	2	2	1	1			2		2			1		3	2		2	
15	1			1	2	2		1			2		2			1			2		2	
16				1																		
17	1		1	1	2	2		1			2		2		1	1		2	2	1	2	
18				1				1								1						
19				1												1						
20	1		2	1	2	2	2	1			2		2		1	1			2		2	
21	1			1	1			1					1			1			1			
22	1			1				1					1		1	1		1	1		2	

Table 1 provides information about which outcomes are A-dominated by other outcomes for a given preference. The 22 Pareto non-dominated outcomes correspond to the rows and columns of the table. An entry $(p,r), p,r = 1,\ldots,22$, in the table may assume a value of $v(p,r) = t, t = 1,2,3$ indicating that the outcome p is A_t-dominated by the outcome r. For each preference, all allowable tradeoffs not specified are assumed to be equal to 10, we note that outcome 10 is not A_t-dominating any other outcome, and is therefore referred to as *weak*. Similarly, based on row 4, we observe that outcome 4 is not A_t-dominated by any other outcome, and is therefore referred to as *strong*. This strong outcome could become a final preferred

Table 2 Dominating and dominated outcomes for tractor–trailer design ($a_{ij} = 0.4$)



design. However, if the designer preferred to obtain a short list of strong outcomes for further consideration rather than one strong outcome, the coefficients $a_{ij} = 1$ should be reduced. In Table 2, that provides the same information as Table 1 but for all $a_{ij} = 0.4$, there are 11 weak outcomes (2, 3, 6, 9, 10, 12, 14, 17, 20, 21, 22) and 5 strong outcomes (4, 5, 13, 18, 19) from which a preferred outcome could be selected as the proposed design. For the application of Model 1 to the same design problem, the reader is referred to [12].

4 Solution Approaches

To develop optimality concepts and related solution approaches to MSMOP we consider another simplified version of this formulation and assume that the preference cones of all MOPs of MSMOP are identical, $C^s = C$ for all $s \in S$. Let $F = [f^1, f^2, \ldots, f^N]$ be the overall vector objective. In this case, the concept of optimality for MSMOP is defined as to find efficient solutions $E(X, F, C)$ in X with respect to the overall vector objective F and the preference cone C.

4.1 All-in-One Approach

The *all-in-one* (AiO) approach, in which MSMOP is converted to a large-scale AiO-MOP of the form

$$(X, F = [f^1, f^2, \ldots, f^N], C), \tag{3}$$

is a natural way to find $E(X,F,C)$. Although this approach seems to be straightforward, it has major drawbacks. The significantly increased number of objective functions makes the physical and geometrical perception of the problem and analyses of tradeoffs between the criteria more difficult. Since a solution that is efficient for AiO-MOP may not be efficient for a single-scenario MOP [17], the lack of efficiency of that solution for the single-scenario problems is not easily controlled.

4.2 Scenario-Oriented Approach

To eliminate the limitations of the AiO approach, and in the future allow for different preference cones among the scenarios, we propose a *scenario-oriented* approach. The original MSMOP is decomposed into a collection of K sub-problems MOP_i, $i = 1,\ldots,K$, with a smaller number $m(i)$ of criteria in each $(m(i) \ll< m(s))$. The collection is given as

$$\{(X, f^i = [f_{i1}(x),\ldots,f_{im(i)}(x)], C), i = 1,\ldots,K\} \tag{4}$$

and $m(i)$ is typically equal to 2, 3, or at most 4. This decomposition should be context or application related, that is, the choice of sub-problems should result from specific features of the problem so that the DM will be able to make trade-off decisions independently for each sub-problem. For example, a criterion may be duplicated in two different sub-problems if trade-offs with the participation of that criterion are of significance.

While the decomposition into sub-problems gives a convenient way to handle and reveal trade-offs within every sub-problem, the trade-off between different sub-problems is to be accomplished by a coordinating mechanism. Without loss of generality, consider a collection of two sub-problems denoted as MOP_i and MOP_j, $i \neq j$. Let x^i be a currently preferred efficient solution of the MOP_i selected by the DM, i.e., $x^i \in E(X, f^i, C)$. If this solution is also in $E(X, f^j, C)$, then it might be the final preferred solution depending on DMs approval. However, in the presence of trade-offs between the sub-problems, this case is very unlikely and the values of criterion functions of MOP_j at x^i may need significant improvement before they are accepted by the DM. To find another feasible solution at which the MOP_js criteria have better values, the DM has to give up some of the performance of the criteria in MOP_i. To coordinate the improvement in MOP_j with the deterioration in MOP_i, the former is modified into the so-called coordination problem COP_j of the form

$$(X(\varepsilon^i), f^j, C). \tag{5}$$

Its feasible set $X(\varepsilon^i)$ is a subset of X and includes additional ε-constraints

$$X(\varepsilon^i) = \{x \in X : f_{i1}(x) \leq f_{i1}(x^i) + \varepsilon_{i1},\ldots,f_{im(i)}(x) \leq f_{im(i)}(x^i) + \varepsilon_{im(i)}\}, \tag{6}$$

where the components of $\varepsilon^i = [\varepsilon_{i1}, \ldots, \varepsilon_{im(i)}] \geq 0$ are tolerances specified by the DM. The following results describe relationships between (weakly) efficient solutions of AiO-MOP, MOP$_i$, and MOP$_j$.

Theorem 3 ([3]).

(a) If $x^* \in E(X(\varepsilon^i), f^j, C)$ *then* $x^* \in \varepsilon$-$E(X, f^i, C)$
(b) If $x^* \in E(X(\varepsilon^i), f^j, C)$ *then* $x^* \in w$-$E(X, F, C)$
(c) If $x^* \in E(X, F, C)$ *then there exists* $\varepsilon^* \geq 0$ *such that* $x^* \in E(X(\varepsilon^*), f^j, C)$

The proofs make use of the concept of epsilon-efficiency [4, 5], which in the literature has accounted for modelling limitations or computational inaccuracies and therefore has been tolerable rather than desirable. Interestingly, the scenario-oriented approach is a method in which epsilon-efficiency is of high significance. We first find ε-efficient solutions of the sub-problems as the efficient solutions of the coordination problems, and then use these ε-efficient solutions to reach every (weakly) efficient solution of AiO-MOP. In other words, we deal with smaller-size sub-problems and generate their solutions but, at the same time, visit and examine (weakly) efficient solutions of AiO-MOP in order to arrive at a final preferred efficient solution of MSMOP.

Based on the above discussion we propose the following interactive procedure to find a solution to MSMOP:

Interactive Procedure for MSMOP

Initialization:

(a) Decompose MSMOP into a collection of K multi-criteria sub-problems MOP$_k$, $k = 1, \ldots, K$.
(b) Find a preferred efficient solution x^i to MOP$_i$ for some $i \in \{1, \ldots, K\}$.
(c) Evaluate criterion values at x^i in all MOP$_k$, $k = 1, \ldots, K$.
(d) If x^i is acceptable for all MOP$_k$, $k = 1, \ldots, K$, then x^i is a preferred solution for MSMOP. Otherwise go to the main step.

Main Step:

(a) Given that x^j is not acceptable for MOPj, $j \in \{1, \ldots, K\}$, specify tolerances ε^i and solve a coordination problem COP$_j$ for x^j, $j \neq i$.
(b) Evaluate criterion values at x^j in all MOP$_k$, $k = 1, \ldots, K$.
(c) If x^j is acceptable for all MOP$_k$, $k = 1, \ldots, K$, then x^j is a preferred solution for MSMOP. Otherwise $i \leftarrow j$ and go to step 1.

Output:

x^j that is acceptable for all MOP$_k$, $k = 1, \ldots, K$, and (weakly) efficient for AiO-MOP.

In the sequentially performed main step of the procedure, the coordination problems solved in Step 1 account for additional tolerances imposed by the DM. This is reflected in ε-constraints added to the feasible set of COP_j to coordinate trade-offs between the sub-problem MOP_j currently examined and the sub-problems coordinated so far.

4.3 Example

We illustrate the interactive procedure on the following bi-scenario bi-objective program (BSBOP):

$$\{(X, f^1 = [f_{11}, f_{12}], C^{\mathrm{Par}}), (X, f^2 = [f_{21}, f_{22}], C^{\mathrm{Par}})\}, \tag{7}$$

where $f_{11}(x_1,x_2) = (x_1-2)^2 + (x_2-1)^2, f_{12}(x_1,x_2) = x_1^2 + (x_2-3)^2, f_{21}(x_1,x_2) = (x_1-1)^2 + (x_2+1)^2, f_{22}(x_1,x_2) = (x_1+1)^2 + (x_2-1)^2$ and $X = \{x \in \mathbb{R}^2 : (x_1^2 - x_2 \leq 0, x_1 + x_2 \leq 2, x-1 \leq 0\}$. We maintain the scenario structure and solve $\mathrm{BOP}_1 = (X, f^1 = [f_{11}, f_{12}], C^{\mathrm{Par}})$ for a preferred $x^1 \in E(X, f^1, C^{\mathrm{Par}})$. Let $x^1 = (x_{11}, x_{12}) = (0.5; 1.5)$ which yields the criterion values $[f_{11}(x^1), f_{12}(x^1)] = [2.5, 2.5]$ and $[f_{21}(x^1), f_{22}(x^1)] = [6.5, 2.5]$. The DM intends to improve the performance of x^1 in BOP2, specifies the tolerances $\varepsilon_{21} = 1$ and $\varepsilon_{22} = 2$, and solves COP_2 = $(X(\varepsilon^2), f^2 = [f_{21}, f_{22}], C^{\mathrm{Par}})$ with the feasible set $X(\varepsilon^2) = \{x \in X : f_{11}(x) \leq f_{11}(x^1) + \varepsilon_{21}, f_{21}(x) \leq f_{21}(x^1) + \varepsilon_{22}\}$ for an x^2, a preferred Pareto efficient solution of COP_2. If x^2 is acceptable for both sub-problems, it is the final preferred solution for the BSBOP. Otherwise, the DM changes the tolerances and solves COP_2 again. Figures 7 and 8 show the objective space of BOP_1 and BOP_2, respectively. Figure 7 (8) depicts the Pareto non-dominated set of BOP_1 (BOP_2) and the image of the Pareto efficient set of BOP_2 (BOP_1), which would not be generated in practice. Each figure also depicts the outcomes generated by the interactive procedure from which the DM might choose a final solution of BSBOP.

All depicted outcomes in both figures are (weakly) non-dominated for AiO-MOP related to BSBOP.

The example illustrates that the procedure results in bringing to the DM's attention certain (weakly) Pareto non-dominated outcomes of AiO-MOP from among all the Pareto non-dominated outcomes of that problem. The outcome of the procedure depends on DMs trade-off decisions made for the sub-problems.

5 Conclusion

In this paper, motivated by many applications in engineering design, we have formalized the concept of *scenario* and proposed multi-scenario multi-objective optimization as a new tool for complex decision making problems with multiple and

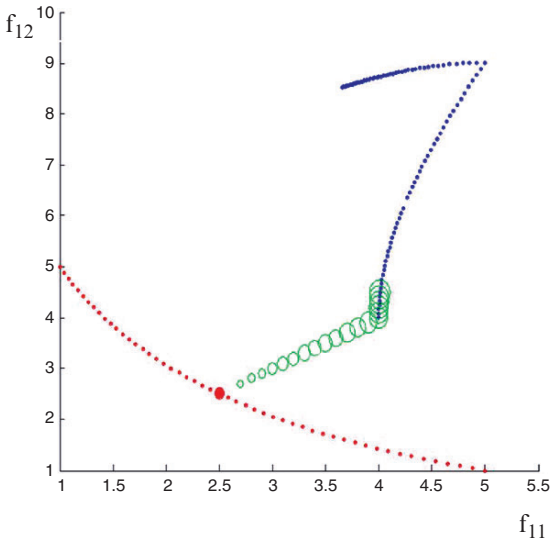


Fig. 7 Non-dominated outcomes of BOP_1 (*lower left curve*); images of efficient solutions of BOP_2 (*upper right curve*); outcomes generated by the procedure (*curve drawn with increasing circles*)

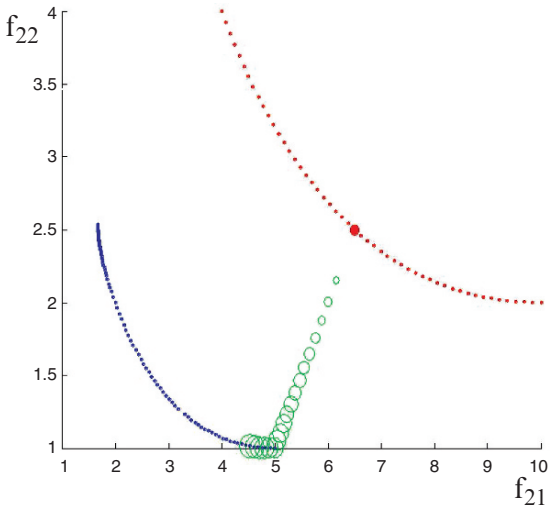


Fig. 8 Non-dominated outcomes of BOP_2 (*lower left curve*); images of efficient solutions of BOP_1 (*upper right curve*); outcomes generated by the procedure (*curve drawn with decreasing circles*)

conflicting criteria. In general, a scenario is understood as a decision situation modelled by an MOP while the overall decision problem of interest requires the consideration of a variety of decision situations or scenarios.

The formulation makes use of a collection of multi-objective programs that, in general, may differ from one another in types and numbers of criterion functions and/or in DMs preferences. We have examined each of these two cases independently of the other, namely, have studied preference modification and implementation for the same criterion vector in every scenario, and have also presented solution approaches to a collection of scenarios with various criterion vectors but one common preference. Since applications have been a driving force for this research, we will continue to apply the models and approaches developed so far to engineering and other real-life problems. Our future work will also encompass the most general case of multiple scenarios with different criterion vectors and preferences.

Acknowledgements This research was partially supported by the Automotive Research Center (ARC), a U.S. Army TACOM Center of Excellence for Modeling and Simulation of Ground Vehicles at the University of Michigan, and by the National Science Foundation grant number DMS-0425768. The views presented here do not necessarily reflect those of our sponsors whose support is gratefully acknowledged.

References

1. Automotive Research Center (1997) ArcSim User Reference Manual. ARC, Ann Arbor, USA
2. El-Gindy M, Woodrooffe J, White D (1991) Evaluation of the dynamic performance of heavy commercial vehicles. In: ASME Annual Meeting, Symposium on Advanced Automotive Technologies, vol DE 40, pp 1–6
3. Engau A (2007) Preferences and tradeoffs in multiobjective programming. Dissertation, Clemson University, South Carolina, USA
4. Engau A, Wiecek M (2007) Exact generation of epsilon-efficient solutions in multiple objective programming. OR Spectrum 29:335–350
5. Engau A, Wiecek M (2007) Generating epsilon-efficient solutions in multiobjective programming. Eur J Oper Res 177:1566–1579
6. Fadel G, Haque I, Blouin V, Wiecek M (2005) Multi-criteria multi-scenario approaches in the design of vehicles. In: Herskovits J, Mazorche S, Canelas A (eds) Proceedings of 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil. Available online at http://www.wcsmo6.org/papers/6391.pdf
7. Fellini R, Kokkolaras M, Papalambros P, Perez-Duarte A (2002) Platform selection under performance loss constraints in optimal design of product families. J Mech Design 127:524–535
8. Geoffrion A (1968) Proper efficiency and the theory of vector maximization. J Math Anal Appl 22:618–630
9. Gobbi M, Mastinu G (2002) Symbolic multiobjective optimization of the dynamic behaviour of actively suspended road vehicles. Int J Vehicle Design 28:189–213
10. Hunt B (2004) Multiobjective programming with convex cones: methodology and applications. Dissertation, Clemson University, South Carolina, USA
11. Hunt B, Wiecek M (2003) Cones to aid decision-making in multicritreia programming. In: Tanino T, Tanaka T, Inuiguchi M (eds) Multi-objective programming and goal programming: theory and applications. Springer, Berlin, pp 153–158
12. Hunt B, Blouin V, Wiecek M (2007) Modeling relative importance of design criteria with a modified Pareto preference. J Mech Design 129:907–914
13. Kouvelis P, Yu G (1997) Robust discrete optimization and its applications. Kluwer, Dordrecht

14. Noghin V (1997) Relative importance of criteria: A quantitative approach. J Multicrit Decision Anal 6:355–363
15. Pareto V (1896) Cours d'conomie politique. Rouge, Lausanne
16. Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, Orlando
17. Singh V (2001) Multi-scenario multi-criteria optimization in engineering design. MS Thesis, Clemson University, South Carolina, USA
18. Sobieszczanski-Sobieski J (1990) Sensitivity analysis and multidisciplinary optimization for aircraft design: recent advances and results. J Aircraft 27:993–1001
19. Wiecek M (2007) Advances in cone-based preference modelling for decision making with multiple criteria. Decision Making Manufact Serv 1:151–171
20. Yu PL (1974) Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. J Optim Theory Appl 14:319–377