Springer Proceedings in Mathematics & Statistics

Athanasios Migdalas
Athanasia Karakitsiou *Editors*

Optimization, Control, and Applications in the Information Age

In Honor of Panos M. Pardalos's 60th Birthday

Springer

Springer Proceedings in Mathematics & Statistics

Volume 130

More information about this series at http://www.springer.com/series/10533

Springer Proceedings in Mathematics & Statistics

This book series features volumes composed of select contributions from workshops and conferences in all areas of current research in mathematics and statistics, including OR and optimization. In addition to an overall evaluation of the interest, scientific quality, and timeliness of each proposal at the hands of the publisher, individual contributions are all refereed to the high quality standards of leading journals in the field. Thus, this series provides the research community with well-edited, authoritative reports on developments in the most exciting areas of mathematical and statistical research today.

Athanasios Migdalas • Athanasia Karakitsiou
Editors

Optimization, Control, and Applications in the Information Age

In Honor of Panos M. Pardalos's 60th Birthday

Springer

Editors
Athanasios Migdalas
Industrial Logistics
ETS Institute
Luleå University of Technology
Luleå, Sweden

Athanasia Karakitsiou
Industrial Logistics
ETS Institute
Luleå University of Technology
Luleå, Sweden

ISSN 2194-1009 ISSN 2194-1017 (electronic)
Springer Proceedings in Mathematics & Statistics
ISBN 978-3-319-18566-8 ISBN 978-3-319-18567-5 (eBook)
DOI 10.1007/978-3-319-18567-5

Library of Congress Control Number: 2015944062

Mathematics Subject Classification (2010): 90C06, 90C08, 90C25, 90C20, 90C26, 90C27, 90C31, 90C35, 90C40, 90C90, 90B35, 90B36, 90B06, 90B20, 46A80, 47H10, 47H09, 93E24, 91D30, 91A10, 91A15, 91A24, 74S05, 49Q10, 15A18.

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Ἄνδρα μοι ἔννεπε, μοῦσα, πολύτροπον, ὃς
μάλα πολλὰ πλάχθη, ἐπεὶ
Τροίης ἱερὸν πτολίεθρον ἔπερσεν,
πολλῶν δ᾽ ἀνθρώπων ἴδεν ἄστεα
καὶ νόον ἔγνω·

(Ὁμήρου Ὀδύσσεια)

Tell me, O Muse, declare to me that man
Tost to and fro by fate, who, when his arms
Had laid Troy's holy city in the dust,
Far wand'ring roam'd on many a tribe of men
To bend his gaze, their minds and thoughts to learn.

(The Odyssey of Homer,
Translated by George Musgrave, 1868)

Σὰ βγεῖς στὸν πηγαμὸ γιὰ τὴν Ἰθάκη
νὰ εὔχεσε νἄναι μακρὺς ὁ δρόμος
γεμάτος περιπέτειες, γεμάτος γνώσεις.
Τοὺς Λαιστρυγόνας καὶ τούς Κύκλωπας,
τὸν θυμωμένο Ποσειδῶνα μὴ φοβᾶσαι,
τέτοια στὸν δρόμο σου ποτὲ σου δὲν θὰ βρεῖς,
ἂν μέν᾽ ἡ σκέψις σου ὑψηλή, ἂν ἐκλεκτὴ
συγκίνησις τὸ πνεῦμα καὶ τὸ σῶμα σου ἀγγίζει.
Τοὺς Λαιστρυγόνας καὶ τοὺς Κύκλωπας,
τὸν ἄγριο Ποσειδῶνα δὲν θὰ συναντήσεις,
ἂν δὲν τοὺς κουβανεῖς μὲς στὴν ψυχή σου,
ἂν ἡ ψυχή σου δὲν τοὺς στήνει ἐμπρὸς σου.

Νὰ εὔχεσαι νἄναι μακρὺς ὁ δρόμος.
Πολλὰ τά καλοκαιρινὰ πρωϊνὰ νὰ εἶναι
ποὺ μὲ τὶ εὐχαρίστησι, μὲ τὶ χαρὰ
θὰ μπαίνεις σὲ λιμένας πρωτοειδωμένους·
νὰ σταματήσεις σ᾽ ἐμπορεῖα Φοινικικά,
καὶ τὲς καλὲς πραγμάτειες ν᾽ ἀποκτήσεις,
σεντέφια καὶ κοράλλια, κεχριμπάρια κ᾽ ἔβενους,
καὶ ἡδονικὰ μυρωδικὰ κάθε λογῆς,
ὅσο μπορεῖς πιὸ ἄφθονα ἡδονικὰ μυρωδικά·
σὲ πόλεις Αἰγυπτιακὲς πολλές νά πᾶς,
νὰ μάθεις καὶ νὰ μάθεις ἀπ᾽ τοὺς σπουδαγμένους.

Πάντα στὸν νοῦ σου νἄχεις τὴν Ἰθάκη.
Τὸ φθάσιμον ἐκεῖ εἶν᾽ ὁ προορισμός σου.
Ἀλλὰ μὴ βιάζεις τὸ ταξεῖδι διόλου.
Καλλίτερα χρόνια πολλὰ νὰ διαρκέσει·
καὶ γέρος πιὰ ν᾽ ἀράξεις στὸ νησί,
πλούσιος μὲ ὅσα κέρδισες στὸν δρόμο,
μὴ προσδοκῶντας πλούτη νὰ σὲ δώσει ἡ Ἰθάκη.

Ἡ Ἰθάκη σ᾽ ἔδωσε τ᾽ ὡραῖο ταξεῖδι.
Χωρὶς αὐτὴν δὲν θἄβγαινες στὸν δρόμο.
Ἄλλα δὲν ἔχει νὰ σὲ δώσει πιά.

Κι ἂν πτωχικὴ τὴν βρεῖς, ἡ Ἰθάκη δὲν σὲ γέλασε.
Ἔτσι σοφὸς πού ἔγινες, μὲ τόση πεῖρα,
ἤδη θὰ τό κατάλαβες ἡ Ἰθάκες τὶ σημαίνουν.

(«Ἰθάκη» τοῦ Κωνσταντίνου Π. Καβάφη, 1863-1933)

When you set out on your journey for Ithaka
pray that the voyage is a long one,
full of adventure, full of knowledge.
The Laestrygonians and the Cyclops,
the angry Poseidon – do not fear of them:
you'll never find things like that on your path,
as long as you keep your thoughts high,
if fine emotions touch your spirit and your body.
The Laestrygonians and the Cyclops,
the fierce Poseidon – you won't encounter them
unless you carry them within your soul,
unless your soul sets them up in front of you.

Pray that the voyage is a long one.
That the summer mornings are many, when,
with such pleasure, with such joy,
you'll enter ports seen for the first time;
may you stop at Phoenician markets
to buy fine merchandise,
mother of pearl and coral, amber and ebony,
and sensual perfume of every kind –
as many sensual perfumes as you can;
and may you visit many Egyptian cities
to learn and learn from their scholars.

Always keep Ithaka in your mind.
Arriving there is your ultimate goal.
But do not hurry the voyage at all.
Better if it lasts for many years,
and to anchor at the island when you are old,
wealthy with all you have gained on the way,
not expecting that Ithaka will offer you riches.

Ithaka gave you the marvelous voyage.
Without her you would not have set out on the road.
She has nothing left to give you now.

And if you find her poor, Ithaka won't have deceived you.
Wise as you have become, so full of experience,
you must have already understood what Ithakas mean.

("Ithaka" by Constantine P. Cavaphes/Kavafis)

Distinguished Professor Panos Pardalos

With our deepest appreciation, the contributors and the editors, we dedicate this volume to the Distinguished Professor Panos M. Pardalos on the occasion of his 60th birthday.

Preface

During June 15–20, 2014, a group of scientists gathered together in a conference on "*Optimization, Control and Applications in the Information Age*" in order to celebrate and honor Panos M. Pardalos on the occasion of his 60th birthday. The meeting took place at the Meliton Hotel of Porto Carras on the middle leg (Sithonia) of the Chalkidiki peninsula in Macedonia, northern Greece, a place of exquisite beauty and one of Panos's favorite places. The conference was organized by Sergiy Butenko and Athanasios Migdalas and was attended by scientists from all over the world. More than 50 members of this "*Panos's club*" presented talks during this event.

This volume is dedicated to Panos M. Pardalos, on the occasion of his 60th birthday. The articles collected in this volume are based on selected talks presented during the conference. Several members of the Panos's club could not attend conference, but have submitted their papers to this volume in order to honor him.

The papers published in this volume cover a wide range of topics and present recent developments and surveys in research fields to which Pardalos has actively contributed and promoted during his career.

In addition, Panos's spouse, Rosemary Bakker, has written a brief biography describing Panos's exciting journey from a pastoral village on the high mountains in Thessaly, central Greece, to a Distinguished Professorship at the University of Florida, that is, Panos's own Odyssey. We therefore dedicate to him the first verses of Homer's Odyssey and Cavaphes' poem "Ithaka" believing that they accurately describe Panos's past, his present, and his future discovery voyages.

We are indebted to Springer publishers and particularly to Razia Amzad for their support in making the publication of this volume possible.

We join our voice with all conference participants, article contributors, and reviewers who made this volume possible in order to wish Panos “Happy Birthday!” and “Chronia Polla!” and in order to express our deepest appreciation to him as a scientist and as a friend.

Luleå, Sweden — Athanasios Migdalas
Luleå, Sweden — Athanasia Karakitsiou
February 2015

Contents

Contributors

Dionne Aleman Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada

G. Anogiannakis Laboratory of Physiology, Medical School, Aristotle University, Thessaloniki, Greece

Rosemary Bakker Gainesville, Florida

Sandra Eriksson Barman Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden

Jeroen Beliën KU Leuven, Brussels, Belgium

Vyacheslav V. Chistyakov Department of Applied Mathematics and Computer Science, and Laboratory of Algorithms and Technologies for Networks Analysis, National Research University Higher School of Economics, Novgorod, Russian Federation

Didem Cinar Department of Industrial Engineering, İstanbul Technical University, İstanbul, Turkey

Center for Applied Optimization, Faculty of Engineering, University of Florida, Gainesville, FL, USA

R. Connors Leeds University, Institute for Transport Studies, Leeds, England

Gareth Davies Cardiff University, Cardiff School of Mathematics, Cardiff, UK

Ioannis C. Demetriou Division of Mathematics and Informatics, Department of Economics, University of Athens, Athens, Greece

A. Evangelou Laboratory of Physiology, University of Ioannina, Medical School, Ioannina, Greece

Wenjuan Fan Department of Information Management and Information Systems, Hefei University of Technology, School of Management, Hefei, China

Key Laboratory of Process Optimization and Intelligent Decision-Making of Ministry of Education, Hefei, China

Dalila B.M.M. Fontes Faculdade de Economia da, Universidade do Porto and LIAAD INESC TEC, Porto, Portugal

Georgia Foutsitzi Technological Educational Institution of Epirus, Preveza, Greece

Jonathan Gillard Cardiff University, Cardiff School of Mathematics, Cardiff, UK

Gražina Gimbutienė Vilnius University, Institute of Mathematics and Informatics, Vilnius, Lithuania

Christos Gogos Technological Educational Institution of Epirus, Preveza, Greece

Alex Golodnikov Institute of Cybernetics of NAS of Ukraine, Kyiv, Ukraine

José Fernando Gonçalves Faculdade de Economia da, Universidade do Porto and LIAAD INESC TEC, Porto, Portugal

Athanasia Karakitsiou Luleå University of Technology, Industrial Logistics, ETS Institute, Luleå, Sweden

Dmitri E. Kvasov Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Universitá della Calabria, Rende (CS), Italy

Software Department, Lobachevsky State University, Nizhni Novgorod, Russia

Torbjörn Larsson Linköping University, Linköping, Sweden

Peter Lindroth Chassis & Vehicle Dynamics, Volvo Group Trucks Technology, Gothenburg, Sweden

Xinbao Liu Department of Information Management and Information Systems, Hefei University of Technology, School of Management, Hefei, China

Key Laboratory of Process Optimization and Intelligent Decision-Making of Ministry of Education, Hefei, China

Dmitrii Lozovanu Institute of Mathematics and Computer Science, Academy of Sciences, Moldova

Magdalene Marinaki Technical University of Crete, School of Production Engineering and Management, Chania, Greece

Yannis Marinakis Technical University of Crete, School of Production Engineering and Management, Chania, Greece

Fred Mayambala Makerere University, Kampala, Uganda

Athanasios Migdalas Luleå University of Technology, Industrial Logistics, ETS Institute, Luleå, Sweden

Department of Civil Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece

Boris Mirkin School of Computer Science and Information Systems, Birkbeck, University of London, London, UK

Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation

José António Oliveira Centro Algoritmi/Departamento de Produçaõ e Sistemas, Universidade do Minho, Braga, Portugal

Michael Orlov Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation

Raghav Pant University of Oxford, Environmental Change Institute, Oxford, UK

Michael Patriksson Department of Mathematical Sciences, Chalmers University of Technology, Gothenburg, Sweden

Jun Pei Department of Information Management and Information Systems, Hefei University of Technology, School of Management, Hefei, China

Department of Industrial and Systems Engineering, Center for Applied Optimization, University of Florida, Gainesville, FL, USA

Stefan Pickl Universität der Bundeswehr München, Institute for Theoretical Computer Science, Mathematics and Operations Research, Neubiberg, München, Germany

M. Poulos Laboratory of Information Technologies, Ionian University, Corfu, Greece

Nils-Hassan Quttineh Linköping University, Linköping, Sweden

Elina Rönnberg Linköping University, Linköping, Sweden

Claes Rydergren Department of Science and Technology, Linköping University, Norrköping, Sweden

B. Say Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada

Yaroslav D. Sergeyev Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Universitá della Calabria, Rende (CS), Italy

Software Department, Lobachevsky State University, Nizhni Novgorod, Russia

Georgios Stavroulakis Department of Production Engineering and Management, Technical University of Crete, Institute of Computational Mechanics and Optimization University Campus, Chania, Greece

V. Stefanidis Laboratory of Physiology, University of Ioannina, Medical School, Ioannina, Greece

Ann-Brith Strömberg Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden

A. Sumalee Department of Civil Engineering, Faculty of Engineering, King Mongkuts Institute of Technology Ladkrabang, Bangkok, Thailand

A. Szatan Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada

Y. Ilker Topcu Department of Industrial Engineering, İstanbul Technical University, İstanbul, Turkey

Theodore B. Trafalis University of Oklahoma, School of Industrial and Systems Engineering, Norman, OK, USA

Stan Uryasev University of Florida, Gainesville, FL, USA

Jorne Van den Bergh KU Leuven, Brussels, Belgium

Mario Ventresca Purdue University, School of Industrial Engineering, West Lafayette, IN, USA

D. Watling Leeds University, Institute for Transport Studies, Leeds, England

Anatoly Zhigljavsky Cardiff University, Cardiff School of Mathematics, Cardiff, UK

Antanas Žilinskas Vilnius University, Institute of Mathematics and Informatics, Vilnius, Lithuania

Greg Zrazhevsky Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Alex Zrazhevsky American Optimal Decisions, Inc., Gainesville, FL, USA

List of Participants

Dionne M. Aleman
aleman@mie.utoronto.ca

Razia Amzad
razia.amzad@springer.com

Ashwin Arulselvan
ashwin.arulselvan@gmail.com

Rosemary Bakker
rosemarybakker@yahoo.com

Mikhail Batsyn
batsyn@yandex.ru

Alexander S. Belenky
abelenky@hse.ru

Roman Belavkin
r.belavkin@mdx.ac.uk

Vladimir Boginksi
boginski@reef.ufl.edu

Sergiy Butenko
butenko@tamu.edu

Marco M Carvalho
mcarvalho@cs.fit.edu

Vyacheslav V. Chistyakov
vchistyakov@hse.ru

Monica Gabriela Cojocaru
mcojocar@uoguelph.ca

C. Demetriou
demetri@econ.uoa.gr

Sandra D. Ekşioğlu
sde47@ise.msstate.edu

Dalila B.M.M. Fontes
fontes@fep.up.pt

Fernando A.C.C. Fontes
faf@fe.up.pt

Mario R. Guarracino
mario.guarracino@cnr.it

Josef Kallrath
josef.kallrath@t-online.de

Valery Kalyagin
vkalyagin@hse.ru

Alla Kammerdiner
alla@nmsu.edu

Athanasia Karakitsiou
athkar@ltu.se

Hamid Khorasani
khorasani.hamid@gmail.com

Alexander Koldanov
alex.koldanov@gmail.com

Pavlo Krokhmal
krokhmal@engineering.uiowa.edu

Dmitri E. Kvasov
ymalave@tamu.edu

Cesar Malave
malave@tamu.edu

Magdalene Marinaki
magda@dssl.tuc.gr

Yannis Marinakis
marinakis@ergasya.tuc.gr

Dmytro Matsypura
ydmytro.matsypura@gmail.com

Athanasios Migdalas
athmig@ltu.se

Boris Mirkin
bmirkin@hse.ru

Nenad Mladenovic
nenad.mladenovic@brunel.ac.uk

Erick Moreno-Centero
ye.moreno@tamu.edu

Alexey Myachin
a_miachin@mail.ru

Anna Nagurney
nagurney@isenberg.umass.edu

Ladimer S. Nagurney
nagurney@hartford.edu

Aleksey Nikolaev
anikolaev@hse.ru

Fivos Panetsos
fivos.panetsos@opt.ucm.es

Panos Pardalos
pardalos@ufl.edu

Stefan Pickl
stefan.pickl@unibw.de

Jose Principe
principe@cnel.ufl.edu

Leonidas Pitsoulis
pitsouli@auth.gr

Alexander Ponomarenko
aponom84@gmail.com

Mahdi Pourakbari
mahdi.pourakbari@gmail.com

Marios Poulos
mpoulos@ionio.gr

Oleg Prokopyev
droleg@pitt.edu

Steffen Rebennack
srebenna@mines.edu

Pablo San Segundo
pablo.sansegundo@upm.es

Yaroslav D. Sergeyev
yaro@si.dimes.unical.it

Stan Uryasev
uryasev@ufl.edu

Luis Nunes Vicente
lnv@mat.uc.pt

Natalia V. Vyssotskaya
nvyssotskaya@mguu.ru

Anatoly Zhigljavsky
zhigljavskyaa@cardiff.ac.uk

Antanas Žilinskas
antanas.zilinskas@mii.vu.lt

Julius Žilinskas
julius.zilinskas@mii.vu.lt

Panos M. Pardalos: A Brief Biography

Rosemary Bakker

Panos Pardalos was born on June 17, 1954 to parents Miltiades and Kalypso in the small mountain village of Mezillo (now Drossato), Greece. He was the first-born child of his parents. This remote village had no doctor, no midwife, so his birth was assisted by a woman fetched from a neighboring village who had experience with birthing. Drossato is located in central Greece, near the Thessaly valley, nestled in the Pindus mountain range. Accessible via a difficult road, its remote location was influential in Panos' childhood. At the time of his birth, the village population was about 400. Panos attended school in the village, where the teacher was boarded at different homes in the village, and taught the first 6 grades in the elementary school. Because of the nature of the school, where all the children were taught in the same classroom, by the time Panos reached second grade, he knew all the lessons through the sixth grade. He was a good student, and was often called to the blackboard to solve the challenging mathematics problems that baffled the other students. His thirst for knowledge was great, but access to books and new material was limited. No television, no radio, no newspaper, in fact the village had no access via roads (only goat paths), no telephone, and no electricity or running water. Panos tells the story of an examiner who came to the school from the government one time to see if the children were learning from the instructor. The examiner posed the question, "Who has travelled and where did you go?" Panos thought for a moment and then raised his hand. Remember the poverty and isolation of the village and consider the answer he gave. "I travelled to Sweden," he announced. The examiner was surprised and said, "How is that possible, Panos?" To which Panos replied, "I travelled there in my imagination!" Thinking outside of the box, even at that young age!

Panos attended school 6 years in the village and then, as no further schooling was available in Drossato, started (high school) in another village, 2 h away on foot from his village. Panos attended the school in Magiro, Petrillo for 2 years walking back and forth every day, and then, as no high school existed anywhere in the mountains, left his home at the age of 14, alone, to attend high school in Volos. Starting high school, he lived for a time with some distant relatives near Volos, and, then, at one point noticing some children at the school who lived in a children's city, decided that he might qualify for help, so, he wrote a letter to the Minister of Education for Greece. Remember that he was 15 and from a poor, remote area of the Greek mainland. Imagine his surprise to receive an answer, a registered letter from the Minister of Education, telling him to report to a high school on the island of Crete. He travelled to Athens for the first time and took a boat for the first time to Iraklion, got off the boat, and took a bus to the high school in Neapolis. He gave the letter to the school officials who were impressed at the letter, decided that he must have very good connections, and enrolled him at the school. The letter entitled him to full support for his high school career, including room and board and even an allowance for clothing! He attended school there for 1 year, and then went back to Volos where he completed his high school studies.

After high school, he took the entrance exams for the university in Greece, passed with high scores and attended Athens University where he received a degree in Mathematics. A friend who was studying in the United States urged him to come for graduate studies, so on August 17, 1977, he left Greece with a suitcase and $200 that his uncle Petros had loaned him and arrived in New York. He obtained a Master's in Computing and Mathematics from Clarkson University in Potsdam, New York. In 1978, he began studies at the University of Minnesota, where he received a Master's in Mathematics and a Ph.D. in Computers and Information Science, working with J. Ben Rosen as his advisor. Panos worked at Penn State University before moving to the University of Florida, where he is currently a Distinguished Professor in the Industrial and Systems Engineering Department. He is married and has one son, Miltiades.

Modular Lipschitzian and Contractive Maps

Vyacheslav V. Chistyakov

Dedicated to Professor Panos Pardalos
on the occasion of his 60th Birthday

Abstract In the context of metric modular spaces, introduced recently by the author, we define the notion of modular Lipschitzian maps between modular spaces, as an extension of the notion of Lipschitzian maps between metric spaces, and address a modular version of Banach's Fixed Point Theorem for modular contractive maps. We show that the assumptions in our fixed point theorem are sharp and that it guarantees the existence of fixed points in cases when Banach's Theorem is inapplicable.

Keywords Modular space • Modular convergence • Lipschitzian map • Fixed point

MSC2010: 46A80, 47H10, 47H09

1 Introduction

The term *modular* in Functional Analysis is an extension of the term *norm* on a linear space. It was introduced by Nakano [19] in 1950. The modern theory of *modular linear spaces* is closely related to the theory of *Orlicz spaces*. Both theories have been extensively developed by the Polish mathematical school from Poznań [16–18, 20] and Russian mathematical school from Voronezh [14] since the end of the 1950s.

V.V. Chistyakov (✉)
Department of Applied Mathematics and Computer Science, and Laboratory of Algorithms and Technologies for Networks Analysis, National Research University Higher School of Economics, 25/12 Bol'shaya Pechërskaya Street, Nizhny Novgorod 603155, Russian Federation
e-mail: vchistyakov@hse.ru; czeslaw@mail.ru

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_1

According to Orlicz [20], a (classical) *modular* on a real linear space X is a functional $\rho : X \to [0,\infty]$ satisfying the following conditions:

- $\rho(0) = 0$;
- given $x \in X$, if $\rho(\alpha x) = 0$ for all $\alpha > 0$, then $x = 0$;
- $\rho(-x) = \rho(x)$ for all $x \in X$;
- $\rho(\alpha x + \beta y) \le \rho(x) + \rho(y)$ for all $x, y \in X$ and $\alpha, \beta \ge 0$ such that $\alpha + \beta = 1$.

If, instead of the inequality on the previous line, ρ satisfies

- $\rho(\alpha x + \beta y) \le \alpha\rho(x) + \beta\rho(y)$,

then the functional ρ is called a *convex modular* on X.

Although the linear structure of X brings some inflexibility, the theories of *modular linear spaces* and *Orlicz spaces* have broad applications [1, 15–17, 21].

In this chapter, we are going to do the following:

1. extend the classical notion of a modular on a *linear space* to the notion of a modular on an *arbitrary* set (Sect. 2);
2. define metrics by means of modulars (Sect. 3);
3. study a new type of convergence—the *modular convergence* (Sect. 3);
4. study modular Lipschitzian maps (Sect. 4);
5. extend Banach's Contraction Principle to the modular context (Sect. 5);
6. present an unusual application of the modular fixed point result (Sect. 6).

This contribution is a slightly extended version of my talk *Lipschitz maps in the modular sense and Banach's contraction principle* given at the International Conference "Optimization, Control and Applications in the Information Age" held in Chalkidiki, Greece, June 15–20, 2014.

2 What Is a Modular?

In what follows, X is a given (nonempty) set.

The idea of a *metric* d on X is of a geometric nature: to any two points $x, y \in X$ a number

$$0 \le d(x,y) < \infty \qquad \text{(the } \textit{distance} \text{ between } x \text{ and } y\text{)}$$

is assigned having the usual three properties: nondegeneracy ($x = y$ if and only if $d(x,y) = 0$), symmetry ($d(x,y) = d(y,x)$), and the triangle inequality ($d(x,y) \le d(x,z) + d(z,y)$ for $z \in X$).

The idea of a *modular* w on X can be naturally interpreted in physical terms: to any time $t > 0$ and two points $x, y \in X$ a quantity

$$0 \le w_t(x,y) \le \infty \qquad \text{(the } \textit{velocity} \text{ between } x \text{ and } y \text{ in time } t\text{)}$$

is assigned satisfying three axioms (see Definition 1). The one-parameter family $w = \{w_t\}_{t>0}$ is a *velocity field* on X.

The axioms of a modular are as follows.

Definition 1 ([5, 7]). A *modular* w on a set X is a one-parameter family $\{w_t\}_{t>0}$ of functions of the form $w_t : X \times X \to [0,\infty]$ satisfying, for all $x,y,z \in X$, the following conditions:

(i) $x = y$ if and only if $w_t(x,y) = 0$ for all $t > 0$;
(ii) $w_t(x,y) = w_t(y,x)$ for all $t > 0$;
(iii) $w_{t+s}(x,y) \le w_t(x,z) + w_s(z,y)$ for all $t > 0$ and $s > 0$.

The modular $w = \{w_t\}_{t>0}$ on X is said to be:

• *strict* if, given $x,y \in X$ with $x \neq y$, we have $w_t(x,y) \neq 0$ for all $t > 0$;
• *convex* if, instead of the inequality in (iii), we have

(iv) $w_{t+s}(x,y) \le \dfrac{t}{t+s}\, w_t(x,z) + \dfrac{s}{t+s}\, w_s(z,y)$.

A few comments on axioms (i)–(iv) are in order.

Axiom (i): two points x and y coincide if no movement is needed in order to get from x to y (the velocity $w_t(x,y)$ is zero for all times $t > 0$).
Axiom (ii): the velocity during the movement from x to y in time t is the same as the velocity in the opposite direction in time t.
Axiom (iii): suppose the movement from x to y is done in two ways as follows: (a) moving straightforward from x to y or (b) passing through a third point $z \in X$. Suppose also that the duration of time is the same in each of the two movements, say, $t+s$. Then the velocity $w_{t+s}(x,y)$ in case (a) does not exceed the sum of the partial velocities $w_t(x,z) + w_s(z,y)$ in case (b). This axiom may be called the *modular triangle inequality* (or *triangle inequality for velocities*).

The strictness of a modular is the strengthening of axiom (i): if the velocity $w_t(x,y)$ is equal to zero for a $t > 0$ (but not necessarily for all $t > 0$), then $x = y$.

Finally, the convexity of w means that, along with w, the family $\{tw_t\}_{t>0}$ is also a modular on X.

Example 1 (Modulars).

1. If (X,d) is a metric space with metric d, then the *canonical modular* on X (the mean velocity) is given by

$$w_t(x,y) = \frac{d(x,y)}{t}, \qquad t > 0, \quad x,y \in X.$$

This modular (i.e., the family $w = \{w_t\}_{t>0}$) is strict and convex.
Moreover, given $0 \le p < \infty$, the strict modular

$$w_t(x,y) = \frac{d(x,y)}{t^p}, \qquad t > 0, \quad x,y \in X,$$

is convex if and only if $p \ge 1$.

2. Modulars w on a metric space (X,d) may look unusual [7]:

$$w_t(x,y)=\begin{cases}\infty & \text{if } t<d(x,y),\\ 0 & \text{if } t\geq d(x,y),\end{cases}\qquad t>0,\quad x,y\in X,$$

is a nonstrict convex modular on X, and

$$w_t(x,y)=\begin{cases}1 & \text{if } t\leq d(x,y),\\ 0 & \text{if } t>d(x,y),\end{cases}\qquad t>0,\quad x,y\in X,$$

is a nonstrict and nonconvex modular on X.

3. Modulars w may look quite usual: given $t>0$ and two sequences of real numbers $x=\{x_n\},y=\{y_n\}\in X=\mathbb{R}^{\mathbb{N}}$, we set

$$w_t(x,y)=\sum_{n=1}^{\infty}\left(\frac{|x_n-y_n|}{t}\right)^p,\quad \text{where}\quad 1\leq p<\infty.$$

This is a strict and convex modular on the sequence set X.
If, in this context, we put

$$w_t(x,y)=\sup_{n\in\mathbb{N}}\left(\frac{|x_n-y_n|}{t}\right)^{1/n},$$

then we get an example of a strict nonconvex modular on X. (In these examples, the set of real numbers $\mathbb{R}$ may be replaced by any metric space.)

4. More examples of modulars can be found in [3–11].
5. Given a real linear space X and a functional $\rho:X\to[0,\infty]$, we set

$$w_t(x,y)=\rho\left(\frac{x-y}{t}\right),\qquad t>0,\quad x,y\in X.$$

Then we have [7, Theorem 3.11]: ρ is a classical (convex) modular on X in the sense of Orlicz if and only if the family $w=\{w_t\}_{t>0}$ is a (convex) modular on X in the sense of axioms (i)–(iv).

Two *main properties* of modulars $w=\{w_t\}_{t>0}$ on X are worth mentioning.

Lemma 1 ([7]). *For any given $x,y\in X$, we have*:

(a) *the function $t\mapsto w_t(x,y)$, mapping $(0,\infty)$ into $[0,\infty]$, is nonincreasing on $(0,\infty)$; moreover, if w is convex, then the function $t\mapsto tw_t(x,y)$ is nonincreasing on $(0,\infty)$;*

(b) *one-sided limits from the right $w_{t+0}(x,y)$ and from the left $w_{t-0}(x,y)$ exist in $[0,\infty]$ (i.e., in the extended sense), and the following inequalities hold*:

$$w_{t+0}(x,y)\leq w_t(x,y)\leq w_{t-0}(x,y).$$

Remark 1. If $w = \{w_t\}_{t>0}$ is a modular on X, then the families $w^+ = \{w_{t+0}\}_{t>0}$ and $w^- = \{w_{t-0}\}_{t>0}$ are also modulars on X, having the same properties as the initial modular w.

3 Modular Spaces

In order to be more specific, we will consider *convex modulars* w on X. Throughout the chapter, we fix an $x^\circ \in X$, called the *center* or *representative* (of a modular space).

Definition 2 ([5, 7]). A *modular space* (around x°) is the set

$$X_w^* \equiv X_w^*(x^\circ) = \{x \in X : w_t(x,x^\circ) < \infty \text{ for some } t = t(x) > 0\}$$

of all elements x from X, which are *reachable* from the center x° with a finite velocity $w_t(x,x^\circ)$ for at least some time $t > 0$.

It is known from [7] (cf. also [6]) that the modular space X_w^* is a *metric space*, whose metric d_w^* (induced by w) is defined implicitly as follows:

$$d_w^*(x,y) = \inf\{t > 0 : w_t(x,y) \le 1\}, \qquad x,y \in X_w^*.$$

Remark 2. If the modular w is nonconvex, then the quantity $d_w^*(x,y)$ may not be (well-)defined as a metric on X_w^* (see Example 2 (item 3)). In this case, the function (cf. [7, 11])

$$d_w^0(x,y) = \inf\{t > 0 : w_t(x,y) \le t\}, \qquad x,y \in X_w^*,$$

is a well-defined *metric* on X_w^*.

Note also [11] that $d_{w^+}^* = d_{w^-}^* = d_w^*$ if w is convex, and $d_{w^+}^0 = d_{w^-}^0 = d_w^0$ if the modular w is nonconvex.

The pair (X_w^*, d_w^*) (and (X_w^*, d_w^0) in the nonconvex case) is called a *metric modular space*.

Example 2 (Metric Modular Spaces). Here we follow the order as in Example 1.

1. If $x^\circ \in X$, $p \ge 1$ and $w_t(x,y) = d(x,y)/t^p$, then

$$X_w^* = X \quad \text{and} \quad d_w^*(x,y) = \inf\{t > 0 : w_t(x,y) \le 1\} = \big(d(x,y)\big)^{1/p}.$$

In particular, if $p = 1$, then we get the canonical modular $w_t(x,y) = d(x,y)/t$, so that $X_w^* = X$ and $d_w^* = d$, and the original metric space (X,d) is restored from the canonical modular.
Now, if $0 \le p < 1$, then $X_w^* = X$ and $d_w^0(x,y) = \big(d(x,y)\big)^{1/(p+1)}$.

2. For the first (convex) modular w, we find $X_w^* = X$ and $d_w^* = d_w^0 = d$. For the second (nonconvex) modular w, we have $X_w^* = X$ and $d_w^0(x,y) = \min\{1, d(x,y)\}$, $x, y \in X$.
3. Setting $x^\circ = 0 \in X = \mathbb{R}^{\mathbb{N}}$ (the zero sequence), for the first (convex) modular w from Example 1(3), we find

$$X_w^* = X_w^*(0) = \left\{ x = \{x_n\} \in X : \sum_{n=1}^{\infty} |x_n|^p < \infty \right\} \qquad (p \geq 1)$$

is the usual set ℓ_p of all p-summable real sequences equipped with the usual (here modular generated) metric

$$d_w^*(x,y) = \left(\sum_{n=1}^{\infty} |x_n - y_n|^p \right)^{1/p}, \qquad x, y \in X_w^* = \ell_p.$$

For the second (nonconvex) modular w from Example 1(3), we have: $X_w^* = X_w^*(0)$ is the set of all sequences $x = \{x_n\} \in \mathbb{R}^{\mathbb{N}}$ such that the sequence $\{t^n x_n\}$ is bounded for some $t > 0$. The metric on X_w^* (in our nonconvex case) is given by

$$d_w^0(x,y) = \sup_{n \in \mathbb{N}} |x_n - y_n|^{1/(n+1)}, \qquad x, y \in X_w^*.$$

On the other hand, it is to be noted that the quantity

$$d_w^*(x,y) = \sup_{n \in \mathbb{N}} |x_n - y_n|, \qquad x, y \in X_w^*,$$

is not a well-defined metric on X_w^*: in fact, the sequence $x = \{n\}_{n=1}^{\infty}$ belongs to X_w^*, but $d_w^*(x,0) = \sup_{n \in \mathbb{N}} n = \infty$.

Modulars w on X give rise to a new type of convergence in X_w^*, which is weaker than the metric d_w^*-convergence. The motivation for it is the following lemma.

Lemma 2 ([11]). *Given a convex modular on X, a sequence $\{x_n\}$ from X_w^* and an element $x \in X_w^*$, we have*

$$\lim_{n \to \infty} d_w^*(x_n, x) = 0 \quad \textit{if and only if} \quad \lim_{n \to \infty} w_t(x_n, x) = 0 \ \textit{for all} \ t > 0.$$

Definition 3 ([11]). A sequence $\{x_n\}$ from X_w^* is said to be *modular convergent* to an element $x \in X_w^*$ if there exists $t_0 > 0$ such that $\lim_{n \to \infty} w_{t_0}(x_n, x) = 0$.

The modular convergence (or w-convergence) of $\{x_n\}$ to x is denoted by $x_n \xrightarrow{w} x$. Any such (nonunique, in general) element x is called a *modular limit* of $\{x_n\}$.

From the references above, the following properties hold for the modular convergence:

- The metric d_w^*-convergence of $\{x_n\}$ to x implies the modular w-convergence of $\{x_n\}$ to x, but not vice versa.
- If w is a strict modular on X, then the modular limit is uniquely determined (if it exists).

Example 3 (Modular Convergence). For all modulars w from Examples 1, 2 (items 1 and 3), the metric d_w^*-convergence of $\{x_n\}$ to x is equivalent to the modular w-convergence. However, for modulars w from Examples 1, 2 (item 2), the metric and modular convergences are *not* equivalent. Actually, in this case, we have: every sequence in X, *bounded* in metric d, is modular w-convergent.

Since, in what follows, we address a modular version of the Banach's Contraction Principle, we need the notion of *modular completeness* of X_w^*, which replaces the notion of completeness of the metric space (X_w^*, d_w^*).

Definition 4 ([11]). Given a modular w on X, the modular space X_w^* is called *modular complete* provided the following condition holds: for each sequence $\{x_n\}$ from X_w^*, which is *modular Cauchy*, i.e.,

$$\lim_{n,m\to\infty} w_{t_0}(x_n, x_m) = 0 \quad \text{for some} \quad t_0 > 0,$$

there exists an $x \in X_w^*$ such that

$$\lim_{n\to\infty} w_{t_0}(x_n, x) = 0. \tag{1}$$

Clearly, for a metric space (X, d) equipped with the canonical modular w, the modular completeness of $X_w^* = X$ is equivalent to the usual metric completeness of X (with respect to $d_w^* = d$).

An example of a modular complete modular space X_w^* will be given in Sect. 6 (for more examples, see [11, Sect. 4.3]).

4 Modular Lipschitzian Maps

Let w be a convex modular on X.

In order to (naturally) introduce the modular version of a Lipschitzian map, we first describe Lipschitzian maps $T : X_w^* \to X_w^*$ with respect to metric d_w^* in terms of the underlying modular w.

Lemma 3 ([11, Theorem 4]). *Let $T : X_w^* \to X_w^*$ and $k > 0$ be a given constant. Then the Lipschitz condition*

$$d_w^*(Tx, Ty) \le k\, d_w^*(x, y) \quad \textit{for all} \quad x, y \in X_w^* \tag{2}$$

is equivalent to the following one: given $x,y \in X_w^*$,

$$w_{kt+0}(Tx,Ty) \le 1 \quad \textit{for all} \quad t>0 \quad \textit{such that} \quad w_t(x,y) \le 1.$$

In particular, if (X,d) is a metric space with metric d and $w_t(x,y) = d(x,y)/t$ is the canonical modular on X, then, for a map $T : X_w^* \to X_w^*$, we have

$$w_{kt+0}(Tx,Ty) = \frac{d(Tx,Ty)}{kt} \le 1 \qquad (x,y \in X)$$

for all $t>0$ such that

$$w_t(x,y) = \frac{d(x,y)}{t} \le 1,$$

and so, setting $t = d(x,y)$ with $x \ne y$, we get

$$d(Tx,Ty) \le k\,d(x,y), \tag{3}$$

which is the usual Lipschitz condition in the metric space X.

Lemma 3 implies the following assertion (global in t): if a map $T : X_w^* \to X_w^*$ and a constant $k>0$ are such that

$$w_{kt}(Tx,Ty) \le w_t(x,y) \quad \text{for all} \quad t>0 \quad \text{and} \quad x,y \in X_w^*,$$

then condition (2) holds, i.e., T is a d_w^*-Lipschitzian map.

This motivates the following definition (local in t).

Definition 5 ([11]). A map $T : X_w^* \to X_w^*$ is said to be *modular Lipschitzian* (or w-Lipschitzian) if there are constants $k>0$ and $t_0>0$ such that

$$w_{kt}(Tx,Ty) \le w_t(x,y) \quad \text{for all} \quad 0<t\le t_0 \quad \text{and} \quad x,y \in X_w^*.$$

Clearly, for a metric space (X,d) with the canonical modular, Definition 5 gives back the usual Lipschitz condition (3).

The *least modular Lipschitz constant* of T is denoted by $k_w(T)$:

$$k_w(T) = \inf\big\{k>0 : \text{there exists } t_0>0 \text{ such that } w_{kt}(Tx,Ty) \le w_t(x,y)$$
$$\text{for all } 0<t\le t_0 \text{ and } x,y \in X_w^*\big\}.$$

Since T is modular Lipschitzian, $k_w(T)$ is well-defined and finite.

As in the case of metric Lipschitzian maps, the following properties of $k_w(T)$ hold.

Theorem 1. *Given two modular Lipschitzian maps T and S, mapping the modular space X_w^* into itself, we have*

(a) $k_w(T\circ S)\le k_w(T)\cdot k_w(S)$, *where $T\circ S$ is the usual composed map of T and S*;
(b) *the following value $k_w^\infty(T)$ exists*:

$$k_w^\infty(T)\equiv\lim_{n\to\infty}\big(k_w(T^n)\big)^{1/n}=\inf_{n\in\mathbb{N}}\big(k_w(T^n)\big)^{1/n}\le k_w(T),$$

where T^n designates the nth iterate of the map T.

Proof. (a) Since $k_w(T)$ and $k_w(S)$ are finite, let $k>k_w(T)$ and $l>k_w(S)$. By the definition of the least modular Lipschitz constants, there exist $t_0>0$ and $s_0>0$ such that

$$w_{kt}(Tx,Ty)\le w_t(x,y)\quad\text{and}\quad w_{ls}(Sx,Sy)\le w_s(x,y)$$

for all $0<t\le t_0$, $0<s\le s_0$, and $x,y\in X_w^*$. Setting $t_1=\min\{t_0/l,s_0\}$, for all $0<t\le t_1$, we have (note that $lt\le t_0$ and $t\le s_0$)

$$\begin{aligned}w_{klt}\big((T\circ S)x,(T\circ S)y\big)&=w_{k(lt)}\big(T(Sx),T(Sy)\big)\\&\le w_{lt}(Sx,Sy)\\&\le w_t(x,y)\quad\text{for all}\quad x,y\in X_w^*.\end{aligned}$$

The definition of $k_w(T\circ S)$ implies $k_w(T\circ S)\le kl$, and it remains to pass to the limits as $k\to k_w(T)$ and $l\to k_w(S)$.

(b) By item (a), we find $k_w(T^{n+m})\le k_w(T^n)\cdot k_w(T^m)$ for all natural numbers n and m.

If $k_w(T^{n_0})=0$ for some natural number n_0, then

$$k_w(T^n)\le k_w(T^{n-n_0})\cdot k_w(T^{n_0})=0\quad\text{for all}\quad n>n_0,$$

and so, assertion (b) follows with $k_w^\infty(T)=0$.

Suppose now that $k_w(T^n)>0$ for all natural n. Setting $a_n=\log k_w(T^n)$, we find $a_{n+m}\le a_n+a_m$ for all $n,m\in\mathbb{N}$, and so, there exists the limit (in the extended sense)

$$a\equiv\lim_{n\to\infty}\frac{a_n}{n}=\inf_{n\in\mathbb{N}}\frac{a_n}{n}\in\{-\infty\}\cup\mathbb{R}.$$

Since the exponential function is continuous, we obtain

$$\lim_{n\to\infty}\big(k_w(T^n)\big)^{1/n}=\inf_{n\in\mathbb{N}}\big(k_w(T^n)\big)^{1/n}=e^a.$$

Finally, inequality $k_w(T^n)\le\big(k_w(T)\big)^n$ implies $k_w^\infty(T)\le k_w(T)$. □

5 Modular Contractions

A map $T : X_w^* \to X_w^*$ is called *modular contractive* (or w-contractive) if it satisfies the conditions of Definition 5 with $0 < k < 1$.

The following modular version of Banach's Contraction Theorem was established in [11, Theorem 6] (see also [9, 10]).

Theorem 2. *Suppose the following conditions hold*:

1. *w is a strict convex modular on a set X;*
2. *the modular space X_w^* is modular complete;*
3. *$T : X_w^* \to X_w^*$ is a modular contractive map;*
4. *for each $t > 0$ there exists $x_t \in X_w^*$ such that $w_t(x_t, Tx_t) < \infty$ (such a map T is called* modular reachable*).*

Then T admits a fixed point: $Tx = x$ for some $x \in X_w^*$.

Moreover, if $w_t(x,y)$ is finite for all $t > 0$ and $x, y \in X_w^*$, then

(a) *condition 4 is redundant;*
(b) *the fixed point of T is uniquely determined;*
(c) *for each $x_0 \in X_w^*$ the sequence of iterations $\{T^n x_0\}$ is modular convergent to the fixed point x.*

(Banach's Theorem is a consequence of Theorem 2: on a complete metric space (X,d) consider the canonical modular $w_t(x,y) = d(x,y)/t$.)

Proof (Sketch). Since T is modular contractive, there exist constants $0 < k < 1$ and $t_0 > 0$ such that $w_{kt}(Tx,Ty) \le w_t(x,y)$ for all $0 < t \le t_0$ and $x, y \in X_w^*$. Setting $\tau_0 = (1-k)t_0$, by condition 4, we find $x_0 = x_{\tau_0} \in X_w^*$ such that $C = w_{\tau_0}(x_0, Tx_0) < \infty$.

Let $x_1 = Tx_0$ and $x_n = Tx_{n-1}$ for all $n \ge 2$.

Let us show that $\{x_n\}$ is a modular Cauchy sequence in X_w^*. Since $k^i \tau_0 < \tau_0 < t_0$ for all $i \in \mathbb{N}$, we have

$$w_{k^i\tau_0}(x_i, x_{i+1}) = w_{k(k^{i-1}\tau_0)}(Tx_{i-1}, Tx_i) \le w_{k^{i-1}\tau_0}(x_{i-1}, x_i)$$
$$\le \cdots \le w_{\tau_0}(x_0, x_1) = C.$$

The convexity of the modular w implies the inequality

$$t w_t(x_m, x_n) \le \sum_{i=m}^{n-1} t_i w_{t_i}(x_i, x_{i+1}), \qquad n > m,$$

where $t_i = k^i \tau_0$ and

$$t \equiv t(m,n) = t_m + t_{m+1} + \cdots + t_{n-1} = k^m \tau_0 \frac{1-k^{n-m}}{1-k},$$

and so,

$$w_t(x_m,x_n) \le \sum_{i=m}^{n-1} \frac{t_i}{t} w_{k^i\tau_0}(x_i,x_{i+1}) \le \frac{1}{t}\left(\sum_{i=m}^{n-1} t_i\right) C = C$$

for all natural $n > m$. Taking into account that $t_0 = \tau_0/(1-k) > t$, by the convexity of w, we get, for $n > m$,

$$w_{t_0}(x_m,x_n) \le \frac{t}{t_0} w_t(x_m,x_n) \le k^m C \to 0 \quad \text{as} \quad m \to \infty,$$

which establishes the modular Cauchy property of the sequence $\{x_n\}$.

By the modular completeness of X_w^*, there exists $x \in X_w^*$ such that equality (1) holds, i.e., $\{x_n\}$ is modular convergent to x, and, by the strictness of w, the modular limit x is unique.

In order to show that $Tx = x$, we note that $Tx_n = x_{n+1}$, and so, by axiom (iii),

$$\begin{aligned} w_{(k+1)t_0}(Tx,x) &\le w_{kt_0}(Tx,Tx_n) + w_{t_0}(x_{n+1},x) \\ &\le w_{t_0}(x,x_n) + w_{t_0}(x_{n+1},x) \to 0 \quad \text{as} \quad n \to \infty, \end{aligned}$$

whence $w_{(k+1)t_0}(Tx,x) = 0$. By the strictness of w, we conclude that $Tx = x$. □

6 An Ad Hoc Application

In [11, Sect. 6], an application of Theorem 2 to the Carathéodory-type ordinary differential equations with the right-hand side from the Orlicz space has been given. However, it was observed [11, Sect. 7.3] that the locality (in t) of Definition 5 was not quite achieved, and so, "an appropriate example is yet to be found."

In this section, we present such an example, which shows at the same time that the assumptions in Theorem 2 are sharp (and the classical Banach's Theorem is inapplicable). Due to the simplicity of the example, all main features of Theorem 2 and its difference with the classical fixed point theorem are clearly emphasized.

Let $X = \mathbb{R}$, $x^\circ = 0$ (the center) and, given $t > 0$ and $x, y \in \mathbb{R}$, we set

$$w_t(x,y) = \begin{cases} 0 & \text{if} \quad t > 0 \text{ and } x = y, \\ \infty & \text{if } 0 < t < 1 \text{ and } x \ne y, \\ \dfrac{|x-y|}{t} & \text{if} \quad t \ge 1 \text{ and } x \ne y. \end{cases}$$

We are going to show that the map $T : \mathbb{R} \to \mathbb{R}$ given by $Tx = 2x$, along with the modular $w = \{w_t\}_{t>0}$, satisfies the assumptions of Theorem 2, and so, T admits a fixed point (clearly, $T(0) = 0$).

Claim 1. *w is a strict convex modular on $X=\mathbb{R}$, and $X_w^*=\mathbb{R}$ is the modular space.*

Proof. (i) Clearly, $w_t(x,x)=0$ for all $t>0$ and $x\in\mathbb{R}$. If $t>0$, and $x,y\in\mathbb{R}$ are such that $x\neq y$, then $w_t(x,y)\neq 0$, and so, w is strict.

(ii) The symmetry property $w_t(x,y)=w_t(y,x)$ is obvious.

(iv) In order to show the convexity of w, we assume, with no loss of generality, that $w_t(x,z)$ and $w_s(z,y)$ are finite ($t,s>0$), $x\neq z$, and $z\neq y$. By the definition of w, we have

$$w_t(x,z)=\frac{|x-z|}{t}\quad\text{and}\quad w_s(z,y)=\frac{|z-y|}{s}$$

with $t\geq 1$ and $s\geq 1$. Since $t+s\geq 2$, once again the definition of w gives, for $x\neq y$,

$$\begin{aligned} w_{t+s}(x,y)=\frac{|x-y|}{t+s}&\leq\frac{t}{t+s}\cdot\frac{|x-z|}{t}+\frac{s}{t+s}\cdot\frac{|z-y|}{s}\\ &=\frac{t}{t+s}\,w_t(x,z)+\frac{s}{t+s}\,w_s(z,y).\end{aligned}$$

The assertion that $X_w^*=\mathbb{R}$ is clear. □

Claim 2. *The metric d_w^* on $X_w^*=\mathbb{R}$, induced by w, is given by*

$$d_w^*(x,y)=\inf\{t>0: w_t(x,y)\leq 1\}=\begin{cases} 0 & \text{if} \quad x=y,\\ 1 & \text{if} \quad 0<|x-y|\leq 1,\\ |x-y| & \text{if} \quad |x-y|>1.\end{cases}$$

Claim 3. *The modular convergence in X_w^* is not equivalent to the metric convergence with respect to metric d_w^*.*

In fact, the sequence $x_n=1/n$ is modular convergent to zero, i.e., $x_n\xrightarrow{w}0$, because if $t_0\geq 1$, then

$$w_{t_0}(x_n,0)=\frac{|x_n-0|}{t_0}=\frac{1}{nt_0}\to 0\quad\text{as}\quad n\to\infty.$$

At the same time, since $0<|x_n-0|\leq 1$, we have $d_w^*(x_n,0)=1$ for all n.

Claim 4. *The modular space $X_w^*=\mathbb{R}$ is modular complete.*

Proof. Suppose $\{x_n\}$ is a modular Cauchy sequence in X_w^*, i.e., for some $t_0>0$, $w_{t_0}(x_n,x_m)\to 0$ as $n,m\to\infty$. It follows that, for each $\varepsilon>0$, there exists $n_0(\varepsilon)\in\mathbb{N}$ such that $w_{t_0}(x_n,x_m)\leq\varepsilon$ for all $n,m\geq n_0(\varepsilon)$. Consider the two possibilities: either $0<t_0<1$ or $t_0\geq 1$.

If $0 < t_0 < 1$, then, setting $\varepsilon = 1$, we have $w_{t_0}(x_n, x_m) \leq 1$, which implies $w_{t_0}(x_n, x_m) = 0$, i.e., $x_n = x_m$, for all $n, m \geq n_0(1)$. Setting $x = x_{n_0(1)}$, we find $x_n = x$, and so, $w_{t_0}(x_n, x) = 0$ for all $n \geq n_0(1)$.

Now, if $t_0 \geq 1$, then, for $\varepsilon > 0$, we get

$$w_{t_0}(x_n, x_m) = \frac{|x_n - x_m|}{t_0} \leq \varepsilon \quad \text{for all} \quad n, m \geq n_0(\varepsilon).$$

Therefore, $\lim_{m\to\infty} x_m = x$ for some $x \in \mathbb{R}$, and so,

$$w_{t_0}(x_n, x) = \frac{|x_n - x|}{t_0} = \lim_{m\to\infty} \frac{|x_n - x_m|}{t_0} \leq \varepsilon$$

for all $n \geq n_0(\varepsilon)$, which means that $\lim_{n\to\infty} w_{t_0}(x_n, x) = 0$. □

Claim 5. *The map $Tx = 2x : \mathbb{R} \to \mathbb{R}$ is modular contractive and its least modular Lipschitz constant is $k_w(T) = 0$, but T is not contractive with respect to metric d_w^*.*

Proof. Let $0 < k < 1$ and $0 < t_0 < 1$ be arbitrary chosen and fixed. Then, by the definition of w, for all $0 < t \leq t_0$ and $x, y \in \mathbb{R}$, we have

$$w_{kt}(Tx, Ty) = w_{kt}(2x, 2y) = \begin{cases} 0 & \text{if } x = y, \\ \infty & \text{if } x \neq y, \end{cases} = w_t(x, y).$$

(This is a very weak and degenerate *velocity contraction*, which is nonintuitive from the point of view of metric contractions!)

By the definition of $k_w(T)$, we find $k_w(T) \leq k$ for all $0 < k < 1$, and so, $k_w(T) = 0$.

If we assume that $d_w^*(Tx, Ty) \leq k\, d_w^*(x, y)$ for some constant $k > 0$, then taking $|x - y| > 1$, we get

$$|2x - 2y| = d_w^*(Tx, Ty) \leq k\, d_w^*(x, y) = k|x - y|,$$

and so, $k \geq 2$, i.e., T is not d_w^*-contractive. □

Remark 3. The contractive condition in Claim 5 is of local character with respect to t: it does not hold for $t_0 = 1$. In fact, if $x \neq y$, $0 < k < 1$, and $t_0 = 1$, then

$$w_{kt_0}(Tx, Ty) = w_k(2x, 2y) = \infty > |x - y| = w_1(x, y) = w_{t_0}(x, y).$$

Claim 6. *The map $Tx = 2x$ is modular reachable in the sense of Theorem 2 (item 4).*

In fact, setting $x_t = 0$ for all $t > 0$, we find

$$w_t(x_t, Tx_t) = w_t(x_t, 2x_t) = w_t(0, 0) = 0.$$

Note that no other point is modular reachable for T: if $x \neq 0$, then $x \neq 2x = Tx$, and so, $w_t(x, Tx) = \infty$ for all $0 < t < 1$.

By Theorem 2, the map $Tx = 2x$ admits a fixed point in $\mathbb{R}$.

Remark 4. The map $S : \mathbb{R} \to \mathbb{R}$ given by $Sx = x+1$ is also modular contractive (with respect to w), but it is not modular reachable: since $x \neq Sx$ for all $x \in \mathbb{R}$ (i.e., S is fixed point free), then $w_t(x, Sx) = \infty$ for all $0 < t < 1$.

Thus, the modular reachability of the map T in Theorem 2 is essential.

Remark 5. Multiple fixed points may exist for a map $T : X_w^* \to X_w^*$ from Theorem 2 if w does not satisfy the assumption in the second half of that theorem (this is the case for our modular w). In fact, denote by $Tx = [x]$ the largest integer, which does not exceed $x \in \mathbb{R}$. Then T is discontinuous at integer points (in the usual sense), and each integer q is its fixed point: $Tq = [q] = q$. The map $T : \mathbb{R} \to \mathbb{R}$ is w-contractive: if $0 < k < 1$ and $0 < t_0 < 1$, then, for all $0 < t \leq t_0$ and $x, y \in \mathbb{R}$, we have

$$w_{kt}(Tx, Ty) = w_{kt}([x], [y]) = \begin{cases} 0 & \text{if } [x] = [y], \\ \infty & \text{if } [x] \neq [y], \end{cases} \leq \begin{cases} 0 & \text{if } x = y, \\ \infty & \text{if } x \neq y, \end{cases} = w_t(x, y).$$

Moreover, T is modular reachable: for each $t > 0$, we may choose (arbitrarily) an integer $q = q_t$, so that $w_t(q_t, Tq_t) = w_t(q_t, [q_t]) = w_t(q_t, q_t) = 0$.

7 Conclusion

The theory of metric modular spaces [3–11], introduced by the author [5] in 2006, extends simultaneously the theory of metric spaces due to Fréchet [12] and Hausdorff [13] on the one hand, and the theory of modular linear spaces of Nakano [19] and Orlicz [20] on the other hand. In this chapter, we have addressed a modular version of the Banach's [2] Fixed Point Theorem, with no reference to the metric notions, and showed that it can produce new fixed points.

The *metric space theory* is "embedded" into the *modular space theory* (as a pre-limit one) via the single *canonical modular* $w(\alpha, x, y) = \alpha d(x, y)$ with parameter $\alpha = 1/t > 0$, and so, the former theory looks as a "linear" theory (in parameter α). The other various dependences on α in general modulars w present broad possibilities in developing Nonlinear Analysis outside the scope of metric and modular linear spaces.

Acknowledgements The author is partially supported by LATNA Laboratory, NRU HSE, RF government grant, ag. 11.G34.31.0057. I express my sincere gratitude to the organizers of the Conference "Optimization, Control and Applications in the Information Age" (Chalkidiki, Greece, June 15–20, 2014), Sergiy Butenko and Athanasios Migdalas, for an exceptionally nice work-and-leisure atmosphere of the event and the linear (nonparallel) order of presentations, which gave a pleasant feeling of completeness in that every talk could have been attended and appreciated.

References

1. Adams, R.A.: Sobolev Spaces. Pure and Applied Mathematics, vol. 65. Academic, New York (1975)
2. Banach, S.: Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. Fundam. Math. **3**, 133–181 (1922)
3. Chistyakov, V.V.: Selections of bounded variation. J. Appl. Anal. **10**(1), 1–82 (2004)
4. Chistyakov, V.V.: Lipschitzian Nemytskii operators in the cones of mappings of bounded Wiener φ-variation. Folia Math. **11**(1), 15–39 (2004)
5. Chistyakov, V.V.: Metric modulars and their application. Dokl. Akad. Nauk **406**(2), 165–168 (2006) (in Russian). English translation: Dokl. Math. **73**(1), 32–35 (2006)
6. Chistyakov, V.V.: Modular metric spaces generated by F-modulars. Folia Math. **15**(1), 3–24 (2008)
7. Chistyakov, V.V.: Modular metric spaces, I: Basic concepts. Nonlinear Anal. **72**(1), 1–14 (2010)
8. Chistyakov, V.V.: Modular metric spaces, II: Application to superposition operators. Nonlinear Anal. **72**(1), 15–30 (2010)
9. Chistyakov, V.V.: A fixed point theorem for contractions in modular metric spaces. e-Print. arXiv: 1112.5561, 1–31 (2011)
10. Chistyakov, V.V.: Fixed points of modular contractive maps. Dokl. Akad. Nauk **445**(3), 274–277 (2012) (in Russian). English translation: Dokl. Math. **86**(1), 515–518 (2012)
11. Chistyakov, V.V.: Modular contractions and their application. In: Models, Algorithms, and Technologies for Network Analysis. Springer Proceedings in Mathematics & Statistics, vol. 32, pp. 65–92. Springer Science + Business Media, New York (2013)
12. Fréchet, M.: Sur quelques points du calcul functionnel. Rend. Circ. Mat. Palermo **22**(1), 1–72 (1906)
13. Hausdorff, F.: Grundzüge der Mengenlehre. Veit and Company, Leipzig (1914)
14. Krasnosel'skiĭ, M.A., Rutickiĭ, J.B.: Convex Functions and Orlicz Spaces. Fizmatgiz, Moscow (1958) (in Russian). English translation: P. Noordhoff Ltd, Groningen (1961)
15. Lindenstrauss, J., Tzafriri, L.: Classical Banach Spaces. II. Function Spaces. Springer, Berlin (1979)
16. Maligranda, L.: Orlicz Spaces and Interpolation. Seminars in Mathematics, vol. 5. Universidade Estadual de Campinas, Campinas SP (1989)
17. Musielak, J.: Orlicz Spaces and Modular Spaces. Lecture Notes in Mathematics, vol. 1034. Springer, Berlin (1983)
18. Musielak, J., Orlicz, W.: On modular spaces. Stud. Math. **18**, 49–65 (1959)
19. Nakano, H.: Modulared Semi-ordered Linear Spaces. Maruzen, Tokyo (1950)
20. Orlicz, W.: A note on modular spaces. I. Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys. **9**, 157–162 (1961). Reprinted in: W. Orlicz, Collected Papers, Part I, II. PWN—Polish Scientific Publishers, Warsaw, 1142–1147 (1988)
21. Rao, M.M., Ren, Z.D.: Applications of Orlicz Spaces. Monographs and Textbooks in Pure Applied Mathematics, vol. 250. Marcel Dekker, New York (2002)

A Taxonomy for the Flexible Job Shop Scheduling Problem

Didem Cinar, Y. Ilker Topcu, and José António Oliveira

Abstract This chapter aims at developing a taxonomic framework to classify the studies on the flexible job shop scheduling problem (FJSP). The FJSP is a generalization of the classical job shop scheduling problem (JSP), which is one of the oldest NP-hard problems. Although various solution methodologies have been developed to obtain good solutions in reasonable time for FSJPs with different objective functions and constraints, no study which systematically reviews the FJSP literature has been encountered. In the proposed taxonomy, the type of study, type of problem, objective, methodology, data characteristics, and benchmarking are the main categories. In order to verify the proposed taxonomy, a variety of papers from the literature are classified. Using this classification, several inferences are drawn and gaps in the FJSP literature are specified. With the proposed taxonomy, the aim is to develop a framework for a broad view of the FJSP literature and construct a basis for future studies.

Keywords Job shop scheduling • Flexible job shop scheduling • Taxonomy • Review

D. Cinar (✉)
Department of Industrial Engineering, İstanbul Technical University, 34367 İstanbul, Turkey

Center for Applied Optimization, Faculty of Engineering, University of Florida, Gainesville, FL 32611, USA
e-mail: cinard@itu.edu.tr

Y.I. Topcu
Department of Industrial Engineering, İstanbul Technical University, 34367 İstanbul, Turkey
e-mail: topcuil@itu.edu.tr

J.A. Oliveira
Centro Algoritmi/Departamento de Produção e Sistemas, Universidade do Minho, 4710-057 Braga, Portugal
e-mail: zan@dps.uminho.pt

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_2

1 Introduction

The job shop scheduling problem (JSP) is one of the oldest NP-hard problems in the scheduling literature. There are a certain number of jobs which need to be scheduled on a certain number of machines. Each job consists of operations which need to be carried out in a predetermined sequence. Each operation can be processed only on one machine, which is known in advance. The aim of the JSP is to find a schedule that minimizes a performance measure. The flexible job shop scheduling problem (FJSP) is more complex and general than the JSP. In the FJSP, each operation can be processed by one of the machines in a given set. Therefore, the FJSP is both an assignment and a scheduling problem.

The JSP has a history very related to that of the FJSP. The evolution of solution techniques used for deterministic JSPs is surveyed by Jain and Meeran in 1999 [1]. Although the first study on the JSP is not explicitly known, it is accepted that the studies began in the 1950s. A polynomial time algorithm developed by Johnson [2] for a two-machine flow shop scheduling problem can be considered as the first algorithm on the JSP. Besides several polynomial time algorithms, basic and efficient heuristics that constituted a basis for classical scheduling theory were developed in the 1950s. During the 1960s, exact algorithms were studied to find an optimal solution for JSPs. The branch and bound algorithm was the most widely used exact method over the years. Because of the limitations on finding the exact solution for many problems, the emphasis shifted to complexity theory during the 1970s and until the mid-1980s. After noticing that JSPs are NP-hard problems, approximation techniques were developed to find good solutions for larger instances in an acceptable amount of time. Although approximation methods do not guarantee an optimal solution, they are efficient in terms of computational time and effective regarding the solution quality. Priority dispatching rules were the earliest approximation algorithms. In the period from 1988 to 1991, innovative approximation algorithms were developed. The shifting bottleneck procedure developed by Adams et al. [3] was the first approximation algorithm proposed in this period. Later, several solution techniques were combined as hybrid methods to decrease their limitations and increase their effectiveness [1].

Although research on the JSP began in the 1950s, the first known FJSP study was carried out in 1990. Brucker and Schlie [4] developed a polynomial time algorithm to solve the FJSP with two jobs. Later on, various approximation techniques were used and miscellaneous hybrid algorithms were developed to improve the performance of the existing techniques. The evolution of the methodologies for JSPs and FJSPs is illustrated in Fig. 1.

This chapter contributes to the literature by developing a taxonomy for the FJSP to create a framework for future research. Reisman et al. [5] statistically reviewed the flowshop scheduling/sequencing research studies. They classified theoretical and applied articles based on research strategies. Their paper provides a quantitative review of the literature on flowshop scheduling/sequencing. Quadt and Kuhn [6] classified the flexible flow line scheduling studies according to objective, solution method, type of machines (identical, uniform, and unrelated), and setup occurrence.

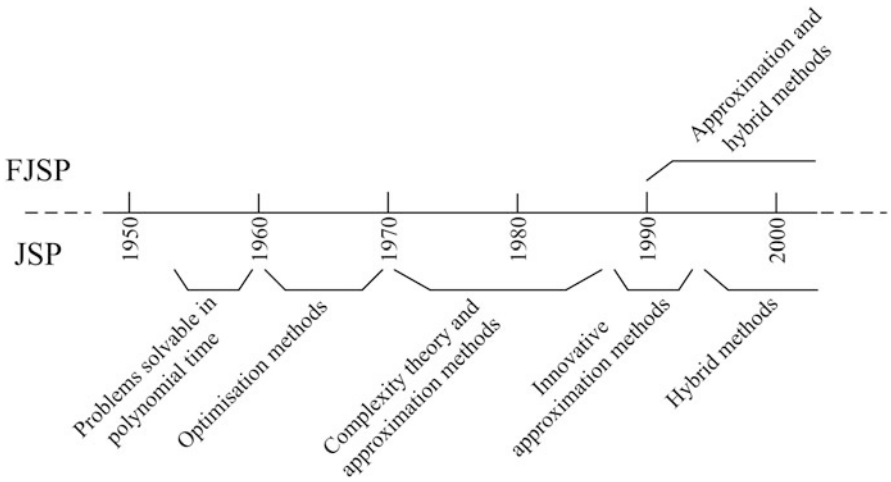


Fig. 1 Historical background for JSP and FJSP

Başar et al. [7] proposed a taxonomic framework for the emergency service station location problem and classified several papers from the literature to verify the proposed framework.

Such a taxonomic framework for the FJSP has not been encountered in the literature review. The present chapter is organized so that the next section gives the explanation of the classical FJSP. Section 3 presents a brief literature review on the FJSP. Section 4 mentions the statistical inferences about the published papers in the FJSP literature. Section 5 clarifies the methodology and defines the proposed taxonomy. The classification of some selected papers and the derived results are given in Sect. 6, whereas the last section is reserved for the conclusion and recommendations.

2 The FJSP

There are n jobs $\{J_1,\ldots,J_n\}$ and m multipurpose machines $\{M_1,\ldots,M_m\}$. Each job J_i consists of n_i operations, $O_{i1},\ldots,O_{in_i}$, which have to be processed in a predetermined sequence without preemption. Operation O_{ij} can be processed on a machine in the given set $\mathfrak{M}_{ij} \subseteq \{M_1,\ldots,M_m\}$. The processing time for operation O_{ij} on machine M_k is denoted by p_{ijk}. A machine cannot process more than one operation at the same time. Furthermore, no two operations of a job can be processed simultaneously, i.e., the predetermined operation sequence of each job has to be enforced. The most widely used performance measure in the literature is makespan (C_{max}), which refers to the longest completion time. The problem is to find a schedule that satisfies both the machine and precedence constraints and minimizes makespan.

The JSP is the special case of the FJSP where $\left|\mathfrak{M}_{ij}\right| = 1$ for all O_{ij}. In the JSP, since the assignment of the operations to machines is predetermined, only the order of the operations is decided. On the other hand, in the FJSP, the aim is finding an assignment and a corresponding schedule that minimize the performance measure [8]. If a feasible assignment is given, then the FJSP becomes a JSP [9].

3 A Brief Literature Review

The FJSP can be divided into two subproblems, the assignment problem and the scheduling problem. For the assignment problem a binary variable V_{ijk} is defined which is 1 if O_{ij} is assigned to machine M_k, otherwise 0. Basically, there are three binary variable definitions for scheduling subproblem.

$$X_{ijkl} = \begin{cases} 1 \text{ if } O_{ij} \text{ is scheduled in the } l\text{th position for processing on } M_k \\ 0 \text{ otherwise} \end{cases}$$

$$Y_{ijkt} = \begin{cases} 1 \text{ if } O_{ij} \text{ is processed by } M_k \text{ during period } t \\ 0 \text{ otherwise} \end{cases}$$

$$Z_{ijghk} = \begin{cases} 1 \text{ if } O_{ij} \text{ precedes } \mathrm{O}_{gh} \text{ (not necessarily immediately) on } M_k \\ 0 \text{ otherwise} \end{cases}$$

They were first proposed by Wagner [10], Bowman [11], and Manne [12], for the JSP. Özgüven et al. [13] developed a mixed integer programming (MIP) model based on Manne's [12] binary variables. Five formulations using the above binary variables were compared by Demir and İşleyen [14] for FJSP in terms of makespan and computation time. The computational tests verified that the formulation having the binary variable proposed by Manne [12] performed better than the others. In 2014, Birgin et al. [15] proposed a novel MIP formulation for the FJPS which is more compact than the one proposed by Özgüven et al. [13] in 2010.

The solution methodologies for deterministic FJSP can be divided into three main categories: exact algorithms, heuristics, and metaheuristics. The branch and bound algorithm, which is one of the exact methods, finds the linear programming (LP) optimum. The performance of this technique depends on the instance and the initial upper bound values [1]. Fattahi et al. [16] used the branch and bound algorithm for the FJSP. Since they could not obtain an optimum solution in a reasonable time, they developed a methodology based on heuristic approaches.

The first developed heuristics to solve classical JSPs are based on priority dispatching rules. Since there is no unique rule which is effective for all problems [17], linear or random combinations of dispatching rules are used to obtain better results. Baykasoğlu and Özbakır [18] analysed the performance of various dispatching rules for the FJSP with different machine flexibilities. Machine flexibility refers to the average number of alternative machines per each operation. Statistical tests showed

that different dispatching rules yield approximately the same performance for the instances with high machine flexibility. Chen et al. [19] proposed a methodology based on priority dispatching rules for a case study in a weapons production factory. Simulation results showed that the proposed model using the combination of earliest due date, the operations' lowest level code of the bill of materials, and the longest processing time outperforms the other scheduling methods. Although priority dispatching rules are easy to implement, the quality of the results decreases when the instance size increases. Because of this drawback and the increasing demand for better solutions, metaheuristic algorithms have been widely used for FJSPs.

Yazdani et al. [20] developed a parallel variable neighborhood search algorithm to solve the FJSP. Parallelization is employed to increase the exploration in the search space by multiple independent searches. Rajkumar et al. [21] developed a GRASP (greedy randomized adaptive search procedure) to solve the multi-objective FJSP with non-fixed availability constraints and compared their results with the results of a hybrid genetic algorithm. Computational results showed that the GRASP algorithm is more appropriate for solving the instances with partial flexibility. Rajkumar et al. [22] also applied a GRASP algorithm to solve the multi-objective FJSP with limited resource constraints.

Saidi-Mehrabad and Fattahi [23] used a tabu search method for the FJSP with sequence dependent set-up times and compared this with the optimal solution obtained by the branch and bound technique. The experimental results showed that the tabu search algorithm achieved optimal solutions in a short computational time for small and medium sized instances. Ennigrou and Ghedira [24] presented two tabu search based multi-agent approaches for the FJSP. The first approach was extended by adding new diversification techniques at both the local and global levels. Because of this diversification, the second approach performs better in terms of makespan. Fattahi et al. [25] applied simulated annealing, which uses stochastic hill climbing procedure to search the solution space, to the FJSP with overlapping.

Genetic algorithms (GA) are among the most widely used approximation algorithms in the FJSP literature. The chromosome representation is an important factor in the performance of a GA [26]. Ho et al. [27] presented a detailed review of chromosome representation for the FJSP. De Giovanni and Pezzella [28] used an operation-based representation in which all the operations of a job are named by the same symbol. The order of occurrence in the given sequence was represented with these symbols. The whole solution space was mapped by an operation-based representation and any permutation of operators can point to a feasible schedule [29]. Mesghouni et al. [30] proposed a parallel job representation for the FJSP. A chromosome was represented by a matrix in which each row refers to the ordered sequence of each job. Saad et al. [31] used this representation to solve a multi-objective FJSP. Parallel job representation needs a repair mechanism after crossover, and the complexity of decoding the representation imposes a high computational cost. Chen et al. [32] proposed an $A-B$ string representation where the A string contains the order of the operations for each job and the B string includes the list of operations that are processed on each machine. The need for a repair

mechanism to maintain feasibility and the high computational cost of checking the consistency between the *A* and *B* strings are the drawbacks of this representation. Kacem et al. [33] proposed an assignment table representation. Since no repair mechanism is required after mutation and crossover, this representation is more effective. Chan et al. [34], Pezzella et al. [8], and Al-Hinai and ElMekkawy [26] used this permutation-based chromosome representation for the FJSP. Moreover, Defersha and Chen [35] implemented this representation for the FJSP with sequence dependent setup times. Ho et al. [27] developed a new chromosomal representation to incorporate a learning mechanism. Only active schedules were generated, so as to reduce the size of the search space. The chromosome consists of two vectors: the operation order part and the machine selection part. The operation order part includes all operations for a job, while the machine assignment vector indicates the machines to which each operation is assigned. Binary values were used in the machine selection component. Moradi et al. [36, 37] used this representation for FJSPs. Zhang et al. [38, 39], Gao et al. [40–42], Li et al. [43, 44], Wang et al. [45], and Xing et al. [46] used similar representations as that of Ho et al. [27], except that the machine selection component was constructed with integer values instead of binary values. Sun et al. [47] used integer values for the operation sequence component that showed the position on the schedule. Frutos et al. [48] used integer values for both components, in which a gene on the operation sequence component represented a possible order of operations on each machines. Jang et al. [49] used the relative operation level besides the machine assignment and operation sequences to determine the operation sequence.

Particle swarm optimization is another evolutionary computation technique. It is inspired by the behavior of a flock of birds. Liu et al. [50] used a multi-particle swarm optimization approach for the multi-objective FJSP. The representation consists of two components: the operation order and the machine selection. According to computational experiments, the proposed algorithm is effective especially for large scale multi-objective FJSP instances. Boukef et al. [51] proposed an algorithm inspired by particle swarm optimization for the FJSP. The computational results confirmed that the efficiency of the proposed algorithm is comparable to the GA in terms of makespan. Pongchairerks and Kachitvichyanukul [52] proposed a new particle swarm optimization approach for the FJSP, where the processing times do not depend on the machines. They implemented multiple social learning topologies in the evolutionary process of particle swarm optimization to avoid being trapped in local optima and to explore various regions in the search space.

Xing et al. [53] proposed a knowledge-based ant colony optimization algorithm, inspired by the behavior of ants. The performance of the proposed method was substantially improved by integrating the ant colony optimization with a knowledge model. Rossi and Dini [54] proposed an ant colony optimization for the FJSP with sequence dependent set-up times. Xing et al. [55] used an ant colony optimization algorithm for the assignment of the operations to the machines and developed a simulation model to solve the multi-objective FJSP. Karthikeyan et al. [56] proposed a firefly algorithm, which is a novel algorithm developed in 2008 to solve nonlinear design problems, for multi-objective FJSPs with limited resource constraints.

Akyol and Bayhan [57] proposed a dynamically coupled neural network for FJSP, in which jobs were not composed of operations. In order to evaluate the performance of the proposed approach, a simulation experiment was performed with different parameters. Bagheri et al. [58] developed an artificial immune algorithm combined with various strategies used for generating the initial population and selecting the individuals for reproduction. Wang and Yu [45] used a filtered beam search algorithm for the FJSP with fixed and non-fixed machine availability constraints. Ziaee [59] developed an efficient heuristic that obtains high quality solutions in a very short computational time. The proposed heuristic can be used to generate an initial solution for metaheuristics in further studies.

In order to improve the quality of the solution, global search algorithms have been frequently hybridized with local search in the FJSP literature. Gao et al. [41] combined GA with a bottleneck shifting procedure to use both the global search ability of GAs and the local search ability of the bottleneck shifting procedure for a multi-objective FJSP. The experimental results showed that the local optima can be improved without too much computational effort, by dynamically adjusting the neighborhood structure. Tay and Ho [60] used composite dispatching rules generated by genetic programming. The computational results verified that no rule performs well on all criteria, and combining the rules increases the efficiency of the procedure and the quality of the results. Zribi et al. [61] solved the FJSP hierarchically as assignment and sequencing subproblems. Two methods based on local search and the branch and bound algorithm are developed for the assignment subproblem, while a hybridized GA is proposed for the sequencing subproblem. Gao et al. [42] combined a two-neighborhood structure with the GA to solve an FJSP with non-fixed availability constraints. Gao et al. [40] used a variable neighborhood search with a GA to improve the search ability. Kacem et al. [33] developed an algorithm called "approach by localization" which is an assignment and scheduling procedure to assign each operation to a suitable machine considering the processing times and workloads of machines. It was inferred that "approach by localization" is more efficient than GA in terms of computational time, and obtained results as good as the results obtained by GA. They also combined GA with the approach by localization to find better results for many real problems. Pezzella et al. [8] applied approach by localization to generate an initial solution for the GA. They also used various dispatching rules to get the sequencing of the initial assignments.

Li et al. [43] hybridized a variable neighborhood search with GAs to solve a multi-objective FJSP. Frutos et al. [48] combined GAs and simulated annealing to integrate a local and global search for solving a multi-objective FJSP. Wang et al. [45] and Zhang et al. [39] proposed a GA based on immune and entropy principles for the multi-objective FJSP. Al-Hinai and ElMekkawy [26] hybridized GA with an initial population generation heuristic and a local search method for the FJSP. Moradi et al. [37] combined GA and priority dispatching rules for FJSP with non-fixed preventive maintenance activities. Xing et al. [46] developed a multi-population interactive coevolutionary algorithm in which both artificial ant colonies and a GA with different configurations were applied to evolve each population independently.

Ho et al. [27] developed a learnable GA which yields an effective integration between evolution and learning within a random search process. Moradi et al. [36] also used a learnable GA to solve the FJSP with preventive maintenance activities.

The particle swarm optimization algorithm is another frequently used approximation method that is combined with other algorithms. Xia and Wu [62] combined simulated annealing with particle swarm optimization for a multi-objective FJSP. Grobler et al. [63] applied four particle swarm optimization-based heuristic approaches to the multi-objective FJSP with sequence-dependent setup times. The priority-based particle swarm optimization algorithm has the best performance in terms of the quality of the solution and the computational complexity. Zhang et al. [64] hybridized a particle swarm optimization algorithm with the tabu search algorithm to solve the multi-objective FJSP. Mosleji and Mahnam [65] combined a particle swarm algorithm and a local search algorithm for multi-objective FJSP with different release times. Li et al. [66, 67] hybridized an artificial bee colony algorithm with the particle swarm methodology and tabu search, for solving the multi-objective FJSP.

Since tabu search is an effective local search algorithm and easy to implement, many methodologies based on tabu search have been developed in the FJSP literature. Scrich et al. [68] developed a hierarchical and multi-start tabu search in which the initial solution is obtained by priority dispatching rules. Fattahi et al. [16] compared integrated and hierarchical approaches for the FJSP and found that the results of the hierarchical algorithms are better than the integrated approaches. According to the experimental results, combining tabu search and simulated annealing algorithms for assignment and sequencing subproblems consecutively had better performance than the other algorithms.

Bozejko et al. [69] used tabu search for the machine selection module and a combination of an insertion algorithm and the tabu search algorithm with backtracking for the operation scheduling module. It was inferred that exact algorithms can be used on both modules to obtain an optimal solution. Li et al. [70] also developed a hybrid algorithm with two modules for multi-objective FJSP. They used a tabu search algorithm to produce neighboring solutions in the machine assignment module and a variable neighborhood search algorithm to apply local search in the operation scheduling component. Li et al. [71] hybridized tabu search with a fast neighborhood structure to solve the FJSP. Wang et al. [72] developed a filtered beam search-based heuristic algorithm to solve the multi-objective FJSP. In order to avoid useless paths and decrease the computational time, heuristics based on dispatching rules were implemented as local and global evaluation functions. Liouane et al. [73] used an ant colony optimization metaheuristic with local search methods including tabu search and showed the efficiency of using local search methods with an ant colony approach.

Although various studies have developed advanced methodologies to solve FJSPs, no review study has been encountered in the scope of this study. This paper proposes a taxonomic framework that can be used to systematically classify the FJPS literature.

4 Statistical Findings

A search was done of the ISI Web of Science using "flexible job shop scheduling," "multipurpose machine job shop," and "job shop scheduling with alternative machines" as the search phrase in the "Subject/Title/Abstract" field options. Only the research papers on deterministic FJSP are included in the statistical analysis. Among the research papers found by this database search, the papers on dynamic FJSPs, lot sizing, or batch splitting in FJSP and rescheduling were eliminated. As a result, 128 research papers are considered in this study.

Initially, the countries of the studies were investigated. The country of the paper is determined by considering the locations of the departments of the authors. For a paper, the authors from the same country are counted once. If the authors are from different countries, then all countries are counted. In this way, countries that are systematically studying FJSPs can be detected. Figure 2 shows the number of studies by country. China is the leading country: one-third of the studies on FJSPs have been carried out there. As is easily seen in Fig. 2, FJSP is mostly studied in eastern countries.

Figure 3 shows the cumulative number of articles with respect to years starting from 1990. Beginning from 2010, a growing interest in FJSP is observed. Table 1 shows the number of articles for each journal where at least two articles have been published. It can be inferred that more than 28 % of the papers have been published by the top two journals in the list.

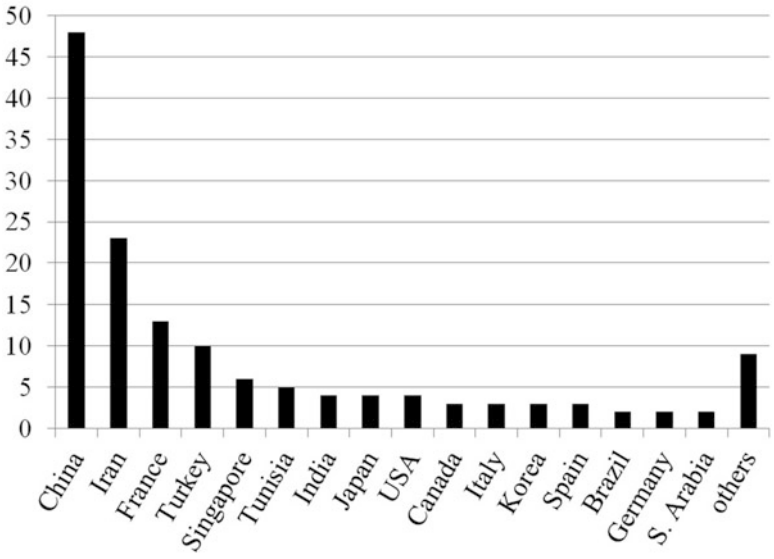


Fig. 2 Number of articles by country

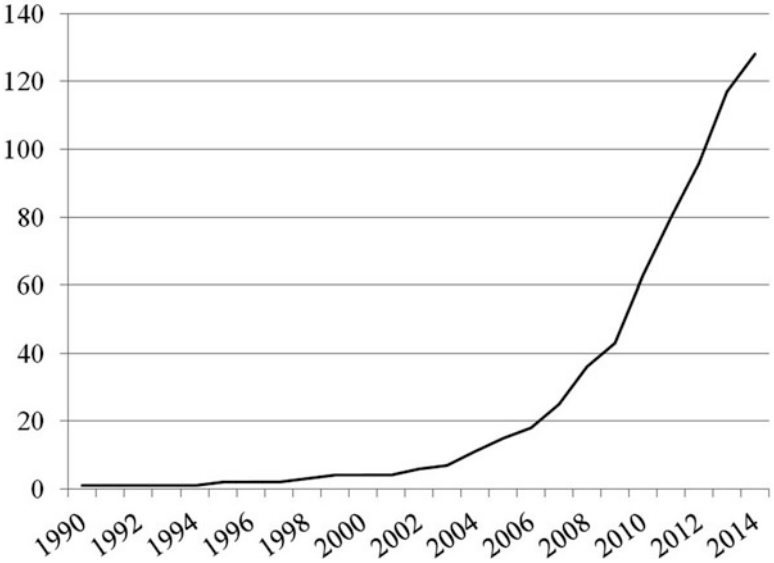


Fig. 3 Cumulative numbers of articles with respect to years

Table 1 Percentage of articles for each journal

International Journal of Production Research	14.84 %
International Journal of Advanced Manufacturing Technology	13.28 %
Computers and Industrial Engineering	7.03 %
Computers and Operations Research	4.69 %
Journal of Intelligent Manufacturing	4.69 %
Applied Mathematical Modelling	3.13 %
Applied Soft Computing	3.13 %
European Journal of Operational Research	3.13 %
Expert Systems with Applications	3.13 %
International Journal of Production Economics	3.13 %
IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews	2.34 %
Advanced Science Letters	1.56 %
Annals of Operations Research	1.56 %
International Journal of Computers Communications & Control	1.56 %
Journal if Manufacturing Systems	1.56 %
Knowledge-Based Systems	1.56 %
Mathematical Problems in Engineering	1.56 %
Studies in Informatics and Control	1.56 %

5 FJSP Taxonomy

In this study, a taxonomic framework is proposed and used for the classification of studies on the static FJSP. Studies on the dynamic FJSP are ruled out. The attribute vector description-based taxonomy method developed by Reisman [74] is used. It has been used for the classification of vehicle routing problem [75] and data envelopment analysis [76].

The attribute vector description-based taxonomy method proceeds in an arborescent way, as shown in Fig. 4. The first level of branching includes the general features of the classified subject in the case of the main topics. They are detailed at the branching levels from top to bottom. At most three branching levels are generated, to yield coherence and parsimony while providing comprehensiveness [75].

The proposed taxonomy for the FJSP literature is given in Fig. 5. The main topics that are placed at the first level of branching are type of study, type of problem, objective, methodology, data characteristics, and benchmarking.

In the first category, a classification is performed with regard to the type of the studies into theory, application, or literature review. Reisman et al. [5] used a classification into theoretical or application for flowshop scheduling/sequencing research studies. According to Reisman et al. [5], "Theoretical papers may be motivated by or even based on real-world problems and offer a wide range of potential applications. Yet, the authors have failed to demonstrate specific examples." They also extended this definition to include papers "that use a previously published scheduling model and proceed to improve the solution technique without adding to the model's real-world validation." According to this definition, reviews and taxonomy studies were considered as theoretical. The terminology used in this study differs from this definition in that reviews and taxonomy studies are handled as a third subcategory besides theoretical and application.

The second category includes the features of the problem and consists of six subcategories: processing time, release time, setup, overlapping, maintenance, and process plan.

Within the first subcategory of problem features, studies are distinguished based on whether the machines are related or unrelated. If machines are related, an operation O_{ij} can be processed at the same time on any machine in the set $\mathfrak{M}_{ij}$. If machines are unrelated, the processing times may be different for each machine.

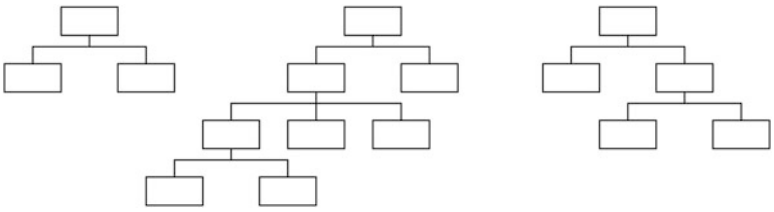


Fig. 4 Attribute vector description-based taxonomy

1. Type of Study
   1.1. Theory
   1.2. Application
   1.3. Review

2. Type of problem
   2.1. Processing time
      2.1.1. Depend on machine
      2.1.2. Do not depend on machine
   2.2. Release time
      2.2.1. Release time for jobs
      2.2.2. Release time for machines
      2.2.3. No release time
   2.3. Set-up
      2.3.1. Sequence dependent set-up
      2.3.2. No set-up
   2.4. Overlapping
      2.4.1. With overlapping
      2.4.2. Without overlapping
   2.5. Maintenance
      2.5.1. Fixed maintenance
      2.5.2. None-fixed maintenance
      2.5.3. Maintenance is not planned

3. Objective
   3.1. Number of objectives
      3.1.1. Single objective
      3.1.2. Multi objective
   3.2. Type of objective function
      3.2.1. Makespan
      3.2.2. Flowtime
      3.2.3. Total workload
      3.2.4. Critical machine workload
      3.2.5. Total tardiness
      3.2.6. Mean tardiness
      3.2.7. Production cost
      3.2.7. Other

4. Methodology
   4.1. Exact method
      4.1.1. Branch and bound algorithm
      4.1.2. Decomposition methods
   4.2. Heuristic
   4.3. Metaheuristic
      4.3.1. Variable neighborhood search
      4.3.2. GRASP
      4.3.3. Tabu search
      4.3.4. Simulated annealing
      4.3.5. Genetic algorithms
      4.3.6. Partical swarm optimization
      4.3.7. Ant colonies
      4.3.8. Neural networks
      4.3.9. Artificial immune system
      4.3.10. Filtered beam search
      4.3.11. Other

5. Data characteristics
   5.1. Data used
      5.1.1. Real world data
      5.1.2. Synthetic data
      5.1.3. No data used
   5.2. Largest instance size
      5.2.1. Number of machines
      5.2.2. Number of jobs
      5.3.3. Total number of operations

6. Benchmark
   6.1. Results of other methods
   6.2. Literature results
   6.3. No benchmark

Fig. 5 Taxonomy of the FJSP literature

In the unrelated case, the processing times depend on the machine. Another subcategory in the second level is release time, which is considered separately for jobs and machines. If release time exists for a job, then the process of that job cannot be initiated before the release time.

Although a setup time is often required between operations in real life, it is not taken into consideration during the modeling of classical scheduling problems, because it makes the problem hard to solve. On the other hand, some studies take into account setup times as sequence dependent or sequence independent. If the setup time is contingent on the immediately preceding operation on the same machine, then it is called a sequence-dependent setup time. Otherwise, the setup time required for each operation is known in advance and is called a sequence-independent setup time [35, 54]. Since sequence-independent setup time can be added to the processing time, it does not need any additional parameter to represent it in the model of the problem. Therefore, a sequence-independent setup time is referred to as "no setup" in the proposed taxonomy.

In classical scheduling problems, no two operations of a job can be processed simultaneously. However, in some FJSPs, an operation of a job can be started before the previous operation is finished because of its nature. The term "overlapping" is used for this feature by Fattahi et al. [25]. Maintenance is another problem feature which can be categorized as fixed and non-fixed maintenance. It is fixed if the starting times of maintenance activities are defined in advance. Flexible starting time for maintenance activities are under consideration in the case of non-fixed maintenance. The third category includes the objectives of the problems. In general, the problems can be classified as either single objective or multi-objective. Some widely used objective functions are shown in Fig. 6. Makespan refers to the longest completion time. Flowtime is the total completion times on all the machines. Total workload is the total working time over all the machines. Critical machine workload is the maximum working time spent on any machine. Total tardiness and mean tardiness are in terms of the due date: tardiness occurs if a job is completed after the due date. Production costs consist of the operating costs, inventory costs, penalty costs for earliness or tardiness, etc. A problem that minimizes one or more of these production costs is assigned to this category. The subcategory "Other" is also added, for the sake of the comprehensiveness of the proposed taxonomy.

The fourth category is reserved for the solution methodologies. This category is divided into three subcategories, depending on the class to which the methodology belongs: exact methods, heuristics, and metaheuristics. Solution techniques in proposed taxonomy are specified by a search of the JSP studies that have been published in recent years. Among these techniques, the branch and bound algorithm and decomposition approaches are the exact methods which guarantee an optimal solution, while the rest are approximation methods. The subcategory "Other" is added to classify the methods that do not belong to the listed techniques.

The data used in the study are the next main category, and are classified based on their origin. According to Eksioglu et al. [75], authors might use real-world data and/or synthetic data that are generated by random number generators or taken from the literature. Since the FJSP is an NP-hard problem, approximation algorithms

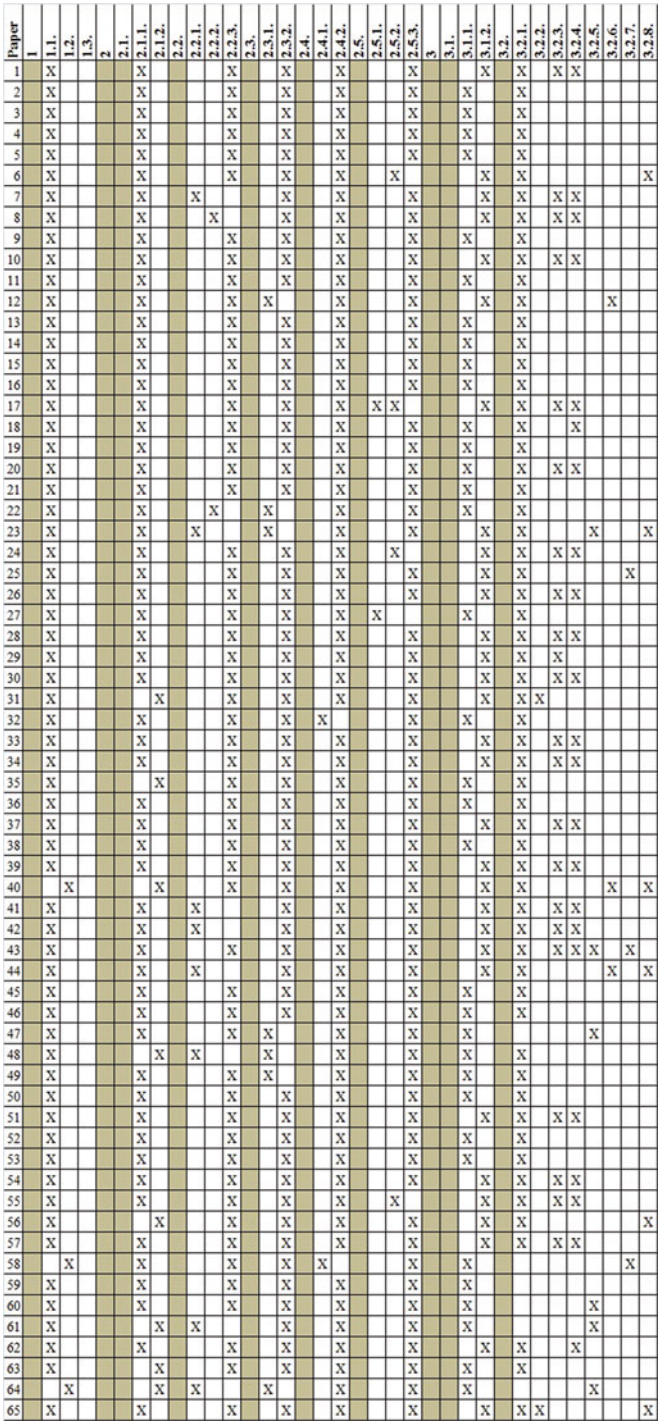


Fig. 6 Classification of selected FJSP papers. (**a**) Classification of FJSP studies with respect to type of study, type of problem, and objective

Paper	4	4.1.	4.1.1.	4.1.2.	4.2.	4.3.	4.3.1.	4.3.2.	4.3.3.	4.3.4.	4.3.5.	4.3.6.	4.3.7.	4.3.8.	4.3.9.	4.3.10.	4.3.11.	5	5.1.	5.1.1.	5.1.2.	5.1.3.	5.2.	5.2.1.	5.2.2.	5.2.3.	6	6.1.	6.2.	6.3.
1					X						X					X					X			20	15	240			X	
2							X														X			100	20	>300		X	X	
3											X										X			20	15	240		X		
4					X						X					X					X			30	15	300			X	
5									X							X					X			20	15	240			X	
6					X						X										X			30	15	300		X		
7												X				X					X			15	10	56			X	
8								X													X			12	5	30		X		
9											X										X			20	15	240			X	
10												X				X					X			15	10	56			X	
11											X		X								X			20	15	240			X	
12							X														X			15	10	150		X		
13											X										X			15	15	225			X	
14													X								X			200	20	<6000			X	
15															X						X			30	15	300			X	
16							X														X			30	15	300			X	
17																X					X			10	10	30			X	
18											X										X			10	10	30			X	
19							X														X			30	15	300			X	
20					X																X			25	10	-			X	
21											X										X			20	4	100		X		
22											X										X			250	40	1583		X		
23					X							X									X			-	-	256		X		
24								X													X			15	10	56		X		
25										X	X										X			15	10	56				X
26							X		X												X			20	15	240			X	
27					X						X										X			30	10	450		X		
28											X				X						X			15	18	225			X	
29											X				X						X			15	10	56			X	
30							X				X										X			15	10	56			X	
31												X									X			15	10	56		X		
32										X											X			12	8	<48		X		
33													X								X			15	10	56		X		
34									X			X									X			15	10	56			X	
35												X									X			15	15	-			X	
36											X		X								X			20	15	240			X	
37																	X				X			20	13	240			X	
38									X												X			30	15	300			X	
39							X				X										X			30	15	300			X	
40					X															X				2	17	112				X
41					X											X					X			20	15	240		X		
42			X		X						X						X				X			30	10	300			X	
43											X										X			15	10	56			X	
44					X						X										X			200	15	-		X		
45												X									X			-	-	-		X		
46					X						X						X				X			30	15	300			X	
47														X							X			8	3	0				X
48													X								X			30	20	<600			X	
49									X												X			15	6	88		X		
50									X				X								X			20	15	240			X	
51					X						X										X			15	10	56			X	
52									X	X											X			12	8	<49		X		
53					X						X										X			200	20	<8000			X	
54			X						X		X										X			70	15	210			X	
55											X						X				X			15	10	56		X		
56											X										X			20	10	100			X	
57										X		X									X			15	10	56			X	
58					X																			-	-	-				
59																						X		2	-	-				X
60					X				X												X			100	10	<1000		X		
61											X										X			30	15	-		X		
62											X						X				X			10	10	30			X	
63					X																	X		-	-	-				X
64											X									X	X			-	2	270-360			X	
65					X												X				X			10	10	100		X		

Fig. 6 (continued) (**b**) Classification of FJSP studies with respect to methodology, data and benchmark

have been used to find good solutions for large size instances in a convenient computational time. Although the efficiency of the algorithms increases depending on technological advances, an optimal solution for many large size instances has still not been found. Therefore, the biggest dimension of the instances solved in a study is crucial for current and future studies. The dimension of the instance is represented by the number of machines, number of jobs, and total number of operations. Quadt and Kuhn [6] also used such a subcategory, called the "largest instance size," for flexible flow line scheduling problems.

The last category is according to the type of benchmarking. In many studies, the results of other studies are used for comparison with the results of the applied methodology to evaluate the computational performance. Benchmarking can also be performed by directly implementing the other solution methods. In this study, sensitivity analyses on the parameters of a proposed method are assigned to the results of other methods.

6 Classification of the FJSP Literature

As mentioned in Sect. 4, 128 papers were encountered during the FJSP search and were taken into account for statistical analysis. In order to verify the proposed taxonomy, 65 randomly selected papers among these 128 FJSP papers are classified using the proposed taxonomic framework. The investigated articles for the taxonomic review are listed in Fig. 6. Each row is related to a research paper, while the columns are for the subcategories. If a study gives no information about a main category, the corresponding cells remain empty for that paper. An empty column specifies that there is no study encountered in the corresponding subcategory. There are two empty columns, which constitute 4.17 % of the subcategories. One of the empty columns is "review," which shows that such a taxonomic framework can be used to fill this gap in the literature. The remaining empty column belongs to solution techniques. "Decomposition approaches" are exact methods which have not been previously applied to FJSPs by anyone.

Subcategories marked only once or twice constitute 2.08 % and 18.8 %, respectively. So, 93.75 % of all subcategories are handled in at least two articles. These percentages verify that the proposed taxonomy is robust enough to clearly and systematically classify the FJSP literature. According to the type of the study, 95.38 % of all the papers are theoretical studies which proposed a solution methodology for various FJSPs. There are only three studies addressing, as an application study, the modeling and solving of a real-life problem as an FJSP. As mentioned before, no review studies were encountered in the scope of this study. A total of 69.23 % of the papers are on the classical FJSP in which all the jobs and machines are ready at the beginning of the planning horizon (no release time), setup times and maintenance times are not planned, and overlapping is not considered. In 87.7 % of the papers, the processing time of an operation is related to the machine. The problems which consider processing times depending on the machine are more complicated than

the ones that do not. 10.77 % of the papers studied FJSPs with sequence-dependent setup times. Maintenance, which makes the problem more complicated to solve, is the least considered problem structure in the FJSP literature.

The percentages of the studies including single objective and multi-objective models are approximately the same. The minimization of the makespan is the most widely used objective function, which occurred in almost all studies (90.77 %). Among the multiobjective studies, total workload and critical machine workload are the most widely used objective functions besides makespan.

In order to find an optimum solution or lower bounds, exact methods can be used. Although the branch and bound technique is encountered in the literature, other exact algorithms, such as column generation, which can also be used to accelerate the branch and bound algorithm, have not been used to solve FJSPs. Since the FJSP is an NP-hard problem, various approximation algorithms have been implemented. GA is the most used approximation technique, while neural networks, GRASP, and filtered beam search are the least. In order to improve the quality of the solution, various search algorithms are frequently hybridized. 47.69 % of the reviewed papers hybridize two or more approximation methods.

There are two theoretical papers referring to "no data," where the objective of the paper is either analyzing the complexity of an FJSP or developing a lower bound. The rest of the theoretical papers proposed a solution method and applied it to data. There are only two application papers using real-world data. According to the data structure, 93.85 % of the papers apply a methodology to synthetic data.

With the exception of application and complexity studies, all the papers compared the results of their proposed methodology with the results of other methodologies or the literature.

7 Concluding Remarks

In this chapter, a taxonomic framework for the FJSP, which is one of the NP-hard problems, is developed. The attribute vector description-based taxonomy method developed by Reisman [74] is used. The main categories used for the classification of the FJSP literature are the type of study, type of problem, objective, methodology, data characteristics, and benchmarking. According to a statistical analysis, a growing interest in FJSP studies is observed since 2010. More than 28 % of the papers have been published in the *International Journal of Production Research* or the *International Journal of Advanced Manufacturing Technology*.

In order to verify the proposed taxonomy, illustrative published papers from the literature are classified. Based on the classification of FJSP studies, the following important inferences and gaps in the FJSP literature can be mentioned:

- Most of the papers (95.38 %) are theoretical papers that propose a methodology for solving an FJSP. The NP-hard structure of the FJSP is an important reason why such a large portion of studies deal with solving these problems.

- The classified studies do not substantially focus on real-life problems. The performance of many developed algorithms tested with synthetic data has not been verified real cases.
- 47.69 % of the reviewed papers hybridize two or more approximation methods. Studies containing both approximation and exact approaches in hybrid methods for FJSPs are rare. Thus, there is an important gap in this area, in which the time efficiency of the approximation methods and the property of convergence in exact methods can be merged.
- Most of the papers (69.23 %) are on the classical FJSP in which setup times, maintenance times, and overlapping are not considered. Some combinations of release time, setup, recirculation, overlapping, and maintenance, which can be observed in real industry problems, have not been encountered.

The main contribution of our study is that it provides a broad review of the FJSP literature and a framework for future studies. The proposed taxonomy can be meaningfully enhanced based on the time, evolution, and content of the subject. For example, as new methodologies are developed and used to solve FJSPs, the "other" sub-category in the methodology can be renamed.

References

1. Jain, A.S., Meeran, S.: Deterministic job-shop scheduling: past, present and future. Eur. J. Oper. Res. **113**(2), 390–434 (1999)
2. Johnson, S.: Optimal two and three stage production schedules with set-up times included. Naval Res. Logist. Q. **1**, 61–68 (1954)
3. Adams, J., Balas, E., Zawack, D.: The shifting bottleneck procedure for job shop scheduling, Manag. Sci. **34**(3), 391–401 (1988)
4. Brucker, P., Schlie, R.: Job-shop scheduling with multipurpose machines. Computing **45**(4), 369–375 (1990)
5. Reisman, A., Kumar, A., Motwani, J.: Flowshop scheduling/sequencing research: a statistical review of the literature, 1952–1994. IEEE Trans. Eng. Manage. **44**(3), 316–329 (1997)
6. Quadt, D., Kuhn, H.: A taxonomy of flexible flow line scheduling procedures. Eur. J. Oper. Res. **178**(3), 686–698 (2007)
7. Başar, A., Çatay, B., Ünlüyurt, T.: A taxonomy for emergency service station location problem. Optim. Lett. **6**(6), 1147–1160 (2012)
8. Pezzella, F., Morganti, G., Ciaschetti, G.: A genetic algorithm for the flexible job-shop scheduling problem. Comput. Oper. Res. **35**(10), 3202–3212 (2008)
9. Hurink, J., Jurisch, B., Thole, M.: Tabu search for the job-shop scheduling problem with multi-purpose machines. OR Spectr. **15**, 205–215 (1994)
10. Wagner, H.M.: An integer linear-programming model for machine scheduling. Naval Res. Logist. Q. **6**(2), 131–140 (1959)
11. Bowman, E.: The scheduling-sequence problem. Oper. Res. **7**, 621–624 (1959)
12. Manne, A.S.: On the job-shop scheduling problem. Oper. Res. **8**(2), 219–223 (1960)
13. Özgüven, C., Özbakır, L., Yavuz, Y.: Mathematical models for job-shop scheduling problems with routing and process plan flexibility. Appl. Math. Model. **34**(6), 1539–1548 (2010)
14. Demir, Y., İşleyen, S.K.: Evaluation of mathematical models for flexible job-shop scheduling problems. Appl. Math. Model. **37**(3), 977–988 (2013)

15. Birgin, E.G., Feofiloff, P., Fernandes, C.G., de Melo, E.L., Oshiro, M.T.I., Ronconi, D.P.: A MILP model for an extended version of the flexible job shop problem. Optim. Lett. **8**(4), 1417–1431 (2014)
16. Fattahi, P., Mehrabad, M.S., Jolai, F.: Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. J. Intell. Manuf. **18**(3), 331–342 (2007)
17. Qi, J.G., Burns, G.R., Harrison, D.K.: The application of parallel multipopulation genetic algorithms to dynamic job-shop scheduling. Int. J. Adv. Manuf. Technol. **16**(8), 609–615 (2000)
18. Baykasoğlu, A., Özbakır, L.: Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system. Int. J. Prod. Econ. **124**(2), 369–381 (2010)
19. Chen, J.C., Chen, K.H., Wu, J.J., Chen, C.W.: A study of the flexible job shop scheduling problem with parallel machines and reentrant process. Int. J. Adv. Manuf. Technol. **39**(3–4), 344–354 (2008)
20. Yazdani, M., Amiri, M., Zandieh, M.: Flexible job-shop scheduling with parallel variable neighborhood search algorithm. Expert Syst. Appl. **37**(1), 678–687 (2010)
21. Rajkumar, M., Asokan, P., Vamsikrishna, V.: A grasp algorithm for flexible job-shop scheduling with maintenance constraints. Int. J. Prod. Res. **48**(22), 6821–6836 (2010)
22. Rajkumar, M., Asokan, P., Anilkumar, N., Page, T.: A grasp algorithm for flexible job-shop scheduling problem with limited resource constraints. Int. J. Prod. Res. **49**(8), 2409–2423 (2011)
23. Saidi-Mehrabad, M., Fattahi, P.: Flexible job shop scheduling with tabu search algorithms. Int. J. Adv. Manuf. Technol. **32**(5–6), 563–570 (2007)
24. Ennigrou, M., Ghedira, K.: New local diversification techniques for flexible job shop scheduling problem with a multi-agent approach. Auton. Agent. Multi-Agent Syst. **17**(2), 270–287 (2008)
25. Fattahi, P., Jolai, F., Arkat, J.: Flexible job shop scheduling with overlapping in operations. Appl. Math. Model. **33**(7), 3076–3087 (2009)
26. Al-Hinai, N., ElMekkawy, T.Y.: An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem. Flex. Serv. Manuf. J. **23**(1), 64–85 (2011)
27. Ho, N.B., Tay, J.C., Lai, E.M.K.: An effective architecture for learning and evolving flexible job-shop schedules. Eur. J. Oper. Res. **179**(2), 316–333 (2007)
28. De Giovanni, L., Pezzella, F.: An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. Eur. J. Oper. Res. **200**(2), 395–408 (2010)
29. Cheng, R., Gen, M., Tsujimura, Y.: A tutorial survey of job-shop scheduling problems using genetic algorithms - I: representation. Comput. Ind. Eng. **30**(4), 983–997 (1996)
30. Mesghouni, K., Hammadi, S., Borne, P.: Evolution programs for job-shop scheduling. In: 1997 IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, Orlando, vol. 1, Oct 1997, pp. 720–725
31. Saad, I., Hammadi, S., Benrejeb, M., Borne, P.: Choquet integral for criteria aggregation in the flexible job-shop scheduling problems. Math. Comput. Simul. **76**(5–6), 447–462 (2008)
32. Chen, H., Ihlow, J., Lehmann, C.: A genetic algorithm for flexible job-shop scheduling. In: Proceedings of 1999 IEEE International Conference on Robotics and Automation, 1999, vol. 2, pp. 1120–1125 (1999)
33. Kacem, I., Hammadi, S., Borne, P.: Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Trans. Syst. Man Cybern. C Appl. Rev. **32**(1), 1–13 (2002)
34. Chan, F.T.S., Wong, T.C., Chan, L.Y.: Flexible job-shop scheduling problem under resource constraints. Int. J. Prod. Res. **44**(11), 2071–2089 (2006)
35. Defersha, F.M., Chen, M.Y.: A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups. Int. J. Adv. Manuf. Technol. **49**(1–4), 263–279 (2010)
36. Moradi, E., Ghomi, S., Zandieh, M.: An efficient architecture for scheduling flexible job-shop with machine availability constraints. Int. J. Adv. Manuf. Technol. **51**(1–4), 325–339 (2010)

37. Moradi, E., Ghomi, S., Zandieh, M.: Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem. Expert Syst. Appl. **38**(6), 7169–7178 (2011)
38. Zhang, G.H., Gao, L., Shi, Y.: An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Syst. Appl. **38**(4), 3563–3573 (2011)
39. Gao, L., Zhang, C.Y., Wang, X.J.: An improved genetic algorithm for multi-objective flexible job-shop scheduling problem. Adv. Mater. Res. **97**, 2449–2454 (2010)
40. Gao, J., Sun, L.Y., Gen, M.S.: A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Comput. Oper. Res. **35**(9), 2892–2907 (2008)
41. Gao, J., Gen, M., Sun, L.Y., Zhao, X.H.: A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. Comput. Ind. Eng. **53**(1), 149–162 (2007)
42. Gao, J., Gen, M., Sun, L.Y.: Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. J. Intell. Manuf. **17**(4), 493–507 (2006)
43. Li, J.Q., Pan, Q.K., Xie, S.X.: A hybrid variable neighborhood search algorithm for solving multi-objective flexible job shop problems. Comput. Sci. Inf. Syst. **7**(4), 907–930 (2010)
44. Lei, D.M.: A genetic algorithm for flexible job shop scheduling with fuzzy processing time. Int. J. Prod. Res. **48**(10), 2995–3013 (2010)
45. Wang, X.J., Gao, L., Zhang, C.Y., Shao, X.Y.: A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. Int. J. Adv. Manuf. Technol. **51**(5–8), 757–767 (2010)
46. Xing, L.N., Chen, Y.W., Yang, K.W.: Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems. Comput. Optim. Appl. **48**(1), 139–155 (2011)
47. Sun, W., Pan, Y., Lu, X.H., Ma, Q.Y.: Research on flexible job-shop scheduling problem based on a modified genetic algorithm. J. Mech. Sci. Technol. **24**(10), 2119–2125 (2010)
48. Frutos, M., Olivera, A.C., Tohme, F.: A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem. Ann. Oper. Res. **181**(1), 745–765 (2010)
49. Jang, Y.J., Kim, K.D., Jang, S.Y., Park, J.: Flexible job shop scheduling with multi-level job structures. JSME Int. J. Ser. C Mech. Syst. Mach. Elem. Manuf. **46**(1), 33–38 (2003)
50. Liu, H.B., Abraham, A., Wang, Z.W.: A multi-swarm approach to multi-objective flexible job-shop scheduling problems. Fundam. Inform. **95**(4), 465–489 (2009)
51. Boukef, H., Benrejeb, M., Borne, P.: Flexible job-shop scheduling problems resolution inspired from particle swarm optimization. Stud. Inf. Control **17**(3), 241–252 (2008)
52. Pongchairerks, P., Kachitvichyanukul, V.: A particle swarm optimization algorithm on job-shop scheduling problems with multi-purpose machines. Asia Pac. J. Oper. Res. **26**(2), 161–184 (2009)
53. Xing, L.N., Chen, Y.W., Wang, P., Zhao, Q.S., Xiong, J.: Knowledge-based ant colony optimization for flexible job shop scheduling problems. Appl. Soft Comput. **10**(3), 888–896 (2010)
54. Rossi, A., Dini, G.: Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robot. Comput. Integr. Manuf. **23**(5), 503–516 (2007)
55. Xing, L.N., Chen, Y.W., Yang, K.W.: Multi-objective flexible job shop schedule: design and evaluation by simulation modeling. Appl. Soft Comput. **9**(1), 362–376 (2009)
56. Karthikeyan, S., Asokan, P., Nickolas, S.: A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints. Int. J. Adv. Manuf. Technol. **72**(9–12), 1567–1579 (2014)
57. Akyol, D.E., Bayhan, G.M.: Multi-machine earliness and tardiness scheduling problem: an interconnected neural network approach. Int. J. Adv. Manuf. Technol. **37**(5–6), 576–588 (2008)
58. Bagheri, A., Zandieh, M., Mahdavi, I., Yazdani, M.: An artificial immune algorithm for the flexible job-shop scheduling problem. Futur. Gener. Comput. Syst. Int. J. Grid Comput. Theory Methods Appl. **26**(4), 533–541 (2010)
59. Ziaee, M.: A heuristic algorithm for solving flexible job shop scheduling problem. Int. J. Adv. Manuf. Technol. **71**(1–4), 519–528 (2014)

60. Tay, J.C., Ho, N.B.: Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Comput. Ind. Eng. **54**(3), 453–473 (2008)
61. Zribi, N., Kacem, I., El Kamel, A., Borne, P.: Assignment and scheduling in flexible job-shops by hierarchical optimization. IEEE Trans. Syst. Man Cybern. C Appl. Rev. **37**(4), 652–661 (2007)
62. Xia, W.J., Wu, Z.M.: An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Comput. Ind. Eng. **48**(2), 409–425 (2005)
63. Grobler, J., Engelbrecht, A.P., Kok, S., Yadavalli, S.: Metaheuristics for the multi-objective FJSP with sequence-dependent set-up times, auxiliary resources and machine down time. Ann. Oper. Res. **180**(1), 165–196 (2010)
64. Zhang, G.H., Shao, X.Y., Li, P.G., Gao, L.: An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Comput. Ind. Eng. **56**(4), 1309–1318 (2009)
65. Moslehi, G., Mahnam, M.: A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. Int. J. Prod. Econ. **129**(1), 14–22 (2011)
66. Li, J.Q., Pan, Q.K., Suganthan, P.N., Chua, T.J.: A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. Int. J. Adv. Manuf. Technol. **52**(5–8), 683–697 (2011)
67. Li, J.-Q., Pan, Q.-K., Tasgetiren, M.F.: A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities. Appl. Math. Model. **38**(3), 1111–1132 (2014)
68. Scrich, C.R., Armentano, V.A., Laguna, M.: Tardiness minimization in a flexible job shop: a tabu search approach. J. Intell. Manuf. **15**(1), 103–115 (2004)
69. Bozejko, W., Uchronski, M., Wodecki, M.: Parallel hybrid metaheuristics for the flexible job shop problem. Comput. Ind. Eng. **59**(2), 323–333 (2010)
70. Li, J.Q., Pan, Q.K., Liang, Y.C.: An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. Comput. Ind. Eng. **59**(4), 647–662 (2010)
71. Li, J., Pan, Q., Xie, S., Wang, S.: A hybrid artificial bee colony algorithm for flexible job shop scheduling problems. Int. J. Comput. Commun. Control **6**(2), 286–296 (2011)
72. Wang, S.J., Zhou, B.H., Xi, L.F.: A filtered-beam-search-based heuristic algorithm for flexible job-shop scheduling problem. Int. J. Prod. Res. **46**(11), 3027–3058 (2008)
73. Liouane, N., Saad, I., Hammadi, S., Borne, P.: Ant systems and local search optimization for flexible job shop scheduling production. Int. J. Comput. Commun. Control **2**(2), 174–184 (2007)
74. Reisman, A.: Management Science Knowledge: Its Creation, Generalization, and Consolidation. Quorum Books, Westport, CT (1992)
75. Eksioglu, B., Vural, A.V., Reisman, A.: The vehicle routing problem: a taxonomic review. Comput. Ind. Eng. **57**(4), 1472–1483 (2009)
76. Gattoufi, S., Oral, M., Reisman, A.: A taxonomy for data envelopment analysis. Socio Econ. Plan. Sci. **38**(2–3), 141–158 (2004)

Sensitivity Analysis of Welfare, Equity, and Acceptability Level of Transport Policies

R. Connors, M. Patriksson, C. Rydergren, A. Sumalee, and D. Watling

Abstract Transport planners face a major challenge to devise policies to meet multiple expectations and objectives. While we know that transport networks are complex, multi-modal, and spatially distributed systems, there is now a long history of mathematical tools which assist planners in understanding travel movements. However, the objectives that they are asked to achieve do not always admit such a quantification, and so there is a potential mismatch between seemingly qualitatively driven objectives and quantitatively expressed models of the transport system. In the present chapter we address this mismatch, by focusing on three objectives that we believe represent the typical interests of a planner. These are namely: is the policy economically justifiable (efficient), is it "fair" (equitable), and is it justifiable to a democratic society (acceptable)? We provide mathematical representations of these three objectives and link them to mathematical theory of transport networks, in which we may explore the sensitivity of travel behaviour (and hence the objectives) to various multi-modal transport policies. The detailed steps for representing the policy objectives and sensitivities in the network are set out, and the results of a case study reported in which road tolls, road capacities, and bus fares are the policy variables. Overall, the chapter sets out a systematic method for planners to choose between multi-modal policies based on these three objectives.

R. Connors • D. Watling
Institute for Transport Studies, Leeds University, Leeds, England
e-mail: rconnors@its.leeds.ac.uk; D.P.Watling@its.leeds.ac.uk

M. Patriksson (✉)
Department of Mathematical Sciences, Chalmers University of Technology,
412 96 Gothenburg, Sweden
e-mail: mipat@chalmers.se

C. Rydergren
Department of Science and Technology, Linköping University, 601 74 Norrköping, Sweden
e-mail: clryd@itn.liu.se

A. Sumalee
Department of Civil Engineering, Faculty of Engineering, King Mongkut's Institute
of Technology Ladkrabang, Bangkok, Thailand
e-mail: asumalee@gmail.com

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_3

Keywords Urban traffic travel mode and route choice modelling • Combined network equilibrium model • Welfare • Equitability and acceptability measures • Entropy • Sensitivity analysis

1 Introduction

Transport planners face a major challenge to devise future transport plans to meet multiple expectations and objectives. In doing so, they must consider the complex nature of the transport system. This complexity derives not only from the multimodal nature of the available transport networks, but also from the diverse spatial distribution of the problems and remedies. It is worth noting the lack of such a spatial dimension in the aggregate economic models commonly used for transport policy analysis. While there may be a justification to neglect it in other areas of population activity, it is surely central to the question of travel, given the indisputable link between the location of transport provision and the centres of economic, employment, education, and residential activity in an urban area. This spatial distinction leads us to consider the distribution of impacts across socio-economic sectors of the population, which are typically not distributed evenly in space across a city. This in turn leads beyond whether a transport policy is economically justifiable (efficient) to issues of whether it is "fair" (equitable) and justifiable to a democratic society (acceptable).

Despite the critical importance of the issue of equity and acceptability of different transport policies, there have not been many researches attempting at formulating a quantifiable measure for these effects. On the other hand, there has been a well-established measure for the social welfare improvement in the system rooted from the economic theory. An advantage for having such a quantifiable measure is that possibility of analysing the impact of different transport policy on these impacts analytically using an appropriate transport model.

This chapter aims to formulate some meaningful indicators for measuring the changes of equity and acceptability as results of different transport policy. In analysing the impact of different transport policy, one of the critical questions is related to the potential benefit/impact of different setting of the policy implementation. For instance, one may be interested in the potential gain in social welfare improvement with different level of public transport fare reductions. Most of this kind of analysis has been carried out simply by testing different levels of policy exhaustively with a transport model. The main problem with doing this is computational time. In addition, the key information needed for the decision maker at this level of decision may be just about the direction and rough level of the magnitude of change of the benefit or impact. This chapter applies the method of sensitivity analysis used in [18, 24, 25] to analyse the sensitivity of each of the outputs mentioned earlier with respect to a small perturbation of different transport policy attributes.

The chapter describes a model for urban traffic travel mode and route choice when there are classes of users having different values of time. The route choice model for private transportation is based on the Wardrop user equilibrium principle, represented through a graph resembling the street network, and the public transportation model is based on a simplified network without direct interaction with the private transportation. The travellers are modelled in homogeneous user groups, where the travellers in a group has an equal value of time. A nested logit type choice model is used for the demand and mode choice. A car and public transport (bus) mode is considered in the model. The combined model includes design parameters such as monetary costs for private and public transportation, public transport frequency, and link capacities on links in the network.

The main purpose of constructing, solving, and analyzing this model is its use for the analysis of a bi-level network design optimization model. Within that model, the lower level will serve the purpose of evaluating the effect of changes in the above-mentioned design parameters upon the decisions made by the travellers. The upper level design objective function is formulated in terms of the design parameters and will be chosen with the end-goal of designing efficient, equitable, and acceptable transport systems. The development of network equilibrium models that may be formulated as optimization problems originates with the seminal contribution of Beckmann et al. [4]. The introduction of variable demand network equilibrium models for one mode, where the destination choice is determined by entropy type distribution models, can be found in the contributions of Florian et al. [13] and Evans [9]. In this work we consider the distribution fixed. The above-mentioned model, referred to as combined trip distribution and assignment models, was extended to two modes and mode choice by Florian and Nguyen [12], where the travel times by two modes, say auto and transit, are not related. These models are convex cost multi-commodity network optimization models that, at the time, were natural to attack by using the Frank–Wolfe algorithm or by the partial linearization method first suggested by Evans [9]. Also the remaining models listed here are of this general form. Florian [11] formulated a two-mode network equilibrium model where the transit travel times depend on the auto travel times, and the auto travel times account for the pressure of the transit vehicles which share the capacity of the roads.

The first formulation of a network equilibrium model with hierarchical logit demand functions is that of Fernandez et al. [10], where the choice of stations for the "park-and-ride" mode is given by a lower nest of the demand function. The solution method suggested is an adaptation of the partial linearization algorithm of Evans [9]. A more recent contribution is that of Abrahamsson and Lundqvist [1], who developed a model for the combined trip distribution, mode choice, and assignment models with hierarchical choices, where distribution may precede mode choice or vice versa. In their model, the transit travel times are independent of the auto travel times. In the model in Florian et al. [14] the trip productions are given by class and purpose of travel. The mode choice model is an aggregated hierarchical logit function [8, p. 219]. There are thirteen classes of travellers, three travel purposes, and multiple modes of travel, which include walk, auto, multiple transit modes, and combined modes ("park-and-ride"). Their solution algorithm is based on a block

Gauss–Seidel decomposition of the model which is akin to the partial linearization method of Evans [9]; in a simplified case of separable cost functions, there exists a convex multicommodity network flow optimization model also for this more general situation. A similar model of Wong et al. [29] also includes a gravity model for the generation of the OD movements; the model is applied to the strategic Hong Kong network, using Evans' algorithm for its solution.

The construction of our nested model is inspired by Fernandez et al. [10], and our method of choice is an adaptation of Evans' partial linearization algorithm.

Equity, like the related concepts of justice, fairness, and right, is not simple to quantify. Different people have different concepts of equity, and the aspects of equity that are deemed important will depend on the particular context and circumstances [20]. There are two dimensions of equity: the vertical and horizontal dimensions. The vertical dimension is related to the inequality of the cost and benefit distributions amongst the different user groups. User groups can be categorized either by socio-economic group (e.g. income level) or need for transport service (e.g. disability). The horizontal equity concerns the distribution of cost and benefit amongst the groups considered as equal. In transport network modelling, the horizontal equity can be linked to the inequality in the cost and benefit of travel between different users from different travel movements (i.e. by OD pair). This can be viewed as the spatial equity measure.

Before defining the measure of equity, the unit of observation for the equity impact must be defined. In the social context, the choice of the unit can be an individual or a collective unit such as a household or a group (e.g. women, the elderly, and the disabled). Of course, the decision upon the unit will be associated with the dimension and context of the inequality measurement. In this chapter, the model formulated earlier allows us to define the unit of observation by the user classes (distinguished by income group or value of time), travel movement, or region. For illustration purpose, we will only observe the inequality amongst travellers from different movements (OD pairs).

In measuring the equity impact, we focus on the inequality of the distribution of the consumer surplus (S). From economic theory, the most useful measure for this purpose is the income distribution index. There are several measures of inequality reflecting different perceptions of inequality. The sets of weights that different views attach to transfers at various points in a distribution are different. This can result in contradictory ranking of a given pair of distributions (see [19]).

The chapter is structured into further five sections. The next section briefly describes the definition of the transport model adopted in this chapter. Then, Sect. 3 explains the formulations of the welfare, equity, and acceptability measures. Section 4 illustrates the application of the sensitivity analysis method with different measures and transport policies. Numerical results are shown in Sect. 5. The network adopted in this test is the network of the city of Norrköping in Sweden. The last section concludes the chapter and suggests further research.

2 Definition of Transport Model

2.1 Notation

We start by introducing the notation and the submodels used for stating the combined network equilibrium model. Let $\mathcal{C}$ be the set of all origin–destination (OD) pairs (p,q) from origin p to destination q. Let $\mathcal{M}$ be the set of user groups. Let d_{pq}^m denote the (given) number of potential travellers between origin p and destination q, with any mode, by travellers in user group m. Let the demand for travellers going by car be denoted by d_{pq}^{cm} and the number of travellers going by bus be denoted by d_{pq}^{bm}. Let $d_{pq}^{dm} := d_{pq}^{cm} + d_{pq}^{bm}$ denote the total number of travellers using the two travel modes.

2.2 The Car (Private Transportation) Network Model

Let the car network be defined by the nodes $\mathcal{N}$ and directed links $\mathcal{L}$. Let $\mathcal{R}_{pq}^m$ denote the nonempty set of simple routes in pair (p,q) for user group $m \in \mathcal{M}$. Denote by h_r^m the flow on route $r \in \mathcal{R}_{pq}$ of users in group $m \in \mathcal{M}$. Let the traffic flow on link l by users in group m be denoted by w_l^m. As a consequence, we must have that

$$v_l = \sum_{m \in \mathcal{M}} w_l^m, \qquad l \in \mathcal{L}. \tag{1}$$

The consistency between route and link flows further requires that

$$w_l^m = \sum_{(p,q) \in \mathcal{C}} \sum_{r \in \mathcal{R}_{pq}^m} \delta_{rl}^m h_r^m, \qquad l \in \mathcal{L}, \quad m \in \mathcal{M}, \tag{2}$$

where the value of the element δ_{rl}^m equals one if link $l \in \mathcal{L}$ is present on route $r \in \mathcal{R}_{pq}^m$, otherwise zero.

The travel time on link l at the flow v_l is assumed to be described by $t_l^c(\rho_l, v_l)$, where t_l^c is differentiable and $t_l^c(\rho_l, \cdot)$ is an increasing and differentiable function on $\mathbf{R}_+$ for every value of $\rho_l \in \mathbf{R}$. Here, ρ_l is a parameter in the link travel time function that is related to the link capacity. The function t_l^c is in our numerical examples given by

$$t_l^c(\rho_l, v_l) = a_l^c + b_l^c \left(\frac{v_l}{\sigma^c (k_l^c + \rho_l)} \right)^{n_l^c}$$

where a_l^c, b_l^c, k_l^c, and n_l^c are positive parameters related to the average travel delay and σ^c is a positive car occupancy parameter. Further, an additional monetary cost τ_l is associated with each link $l \in \mathcal{L}$. Monetary costs for user group m is

transformed into a generalized time by the use of a time parameter for each user group, denoted β^m. A generalized link travel time for user group m is therefore given by $t_l(\rho_l, v_l) + \dfrac{\tau_l}{\beta^m}$.

Based on the above, the route costs for a given user group are given by

$$c_r^{cm} := \sum_{l \in \mathcal{L}} \delta_{rl}^m \left(t_l^c(\rho_l, v_l) + \frac{\tau_l}{\beta^m} \right), \qquad r \in \mathcal{R}_{pq}^m, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}. \tag{3}$$

Given a demand d_{pq}^{cm} for each OD pair and user group, the route choice is modelled through the Wardrop user equilibrium principle. In our notation and for the triple (h, w, v) of flow entities, it is simply stated as the combination of the consistency conditions (1), (2), and the following:

$$h_r^m (c_r^{cm} - \pi_{pq}^{cm}) = 0, \qquad r \in \mathcal{R}_{pq}^m, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \tag{4a}$$

$$c_r^{cm} - \pi_{pq}^{cm} \geq 0, \qquad r \in \mathcal{R}_{pq}, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \tag{4b}$$

$$\sum_{r \in \mathcal{R}_{pq}^m} h_r^m = d_{pq}^{cm}, \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \tag{4c}$$

$$h_r^m \geq 0, \qquad r \in \mathcal{R}_{pq}^m, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}. \tag{4d}$$

The entities π_{pq}^{cm} introduced in (4a–4d) should of course be interpreted as the minimal, that is, equilibrium, cost of the routes utilized in each OD pair (p,q) and for each of the user groups $m \in \mathcal{M}$. Also, the notation c_r^{cm} is as defined in (3).

2.3 The Public Transport Model

The public transport (bus) mode is modelled by one direct link from each origin node to each destination node. No direct interaction between the car and the public transport flows is modelled. The travel time from origin p to destination q using public transport is given by

$$t_{pq}^b(\alpha_{pq}, \rho_{pq}, d_{pq}^b) = a_{pq}^b + \rho_{pq}^b + b_{pq}^b \left(\frac{d_{pq}^b}{2\alpha_{pq} k_{pq}^b} \right)^{n_{pq}^c},$$

where a_{pq}^b and b_{pq}^b are positive parameters related to the average travel delay, k_{pq}^b is a parameter related to the capacity, and α_{pq} is a design parameter related to the bus frequency. The design parameter ρ_{pq} is related to the travel time for the public transport mode in pair (p,q). The travel time is assumed independent of the number of travellers. Without affecting the theoretical and computational properties of the model it is however possible to add to the travel time a function, convex in the number of public transport travellers, that reflects crowding effects in the public transport vehicles.

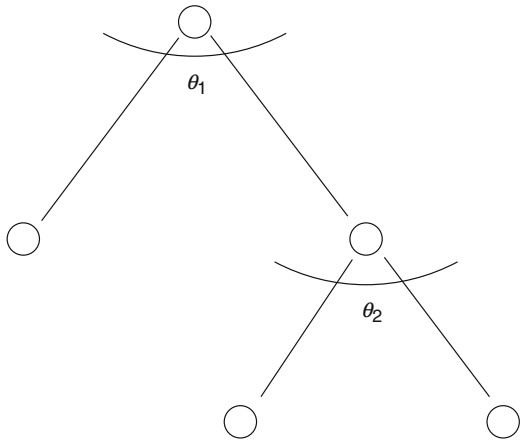


Fig. 1 Nesting of trip and mode choice

The generalized travel time of using the public transport route (or, rather, link) is for each user group and OD pair computed as $c_{pq}^{bm} := t_{pq}^{b} + \dfrac{\tau_{pq}}{\beta^m}$, where τ_{pq} is a positive parameter related to the public transport fare.

2.4 The Travel Demand Model

The travel demand and mode choice is modelled as a nested logit model. The logit model for make-trip versus no-trip is modelled at the first level; for the travellers making the trip, the choice between car and public transport (bus) is modelled at the second level. The construction of the nested model is inspired by Fernandez et al. [10], where a nested model for mode choice and transfer point is presented. The nested logit model is illustrated in Fig. 1.

The parameter θ_1 denotes the dispersion, or uncertainty, in the logit model for the make-trip or no-trip choice and θ_2 is the corresponding dispersion parameter for the bus or car choice on the second level.

The nested logit model can be described in the following form:

$$d_{pq}^{nm} = d_{pq}^{m} \frac{e^{-\theta_1 t_{pq}^{nm}}}{e^{-\theta_1 t_{pq}^{nm}} + e^{-\theta_1 t_{pq}^{dm}}}, \qquad d_{pq}^{dm} = d_{pq}^{m} \frac{e^{-\theta_1 t_{pq}^{dm}}}{e^{-\theta_1 t_{pq}^{nm}} + e^{-\theta_1 t_{pq}^{dm}}}, \tag{5a}$$

$$d_{pq}^{bm} = d_{pq}^{dm} \frac{e^{-\theta_2 t_{pq}^{bm}}}{e^{-\theta_2 t_{pq}^{bm}} + e^{-\theta_2 \pi_{pq}^{cm}}}, \qquad d_{pq}^{cm} = d_{pq}^{dm} \frac{e^{-\theta_2 \pi_{pq}^{cm}}}{e^{-\theta_2 t_{pq}^{bm}} + e^{-\theta_2 \pi_{pq}^{cm}}}, \tag{5b}$$

where t_{pq}^{dm} is the composite cost

$$t_{pq}^{dm} = -\frac{1}{\theta_2} \ln\left(e^{-\theta_2 \pi_{pq}^{cm}} + e^{-\theta_2 t_{pq}^{bm}}\right). \tag{5c}$$

2.5 The Optimization Problem

Let, for short, $x = (h^{\mathrm{T}}, w^{\mathrm{T}}, v^{\mathrm{T}}, (d^b)^{\mathrm{T}}, (d^c)^{\mathrm{T}}, (d^n)^{\mathrm{T}})^{\mathrm{T}}$. The combined model developed above can be stated and solved as the optimization problem to

$$\begin{aligned}
\text{minimize}\, \phi(x) := & \sum_{l\in\mathcal{L}} \left(\int_0^{v_l} t_l^c(\rho_l, s)ds + \sum_{m\in\mathcal{M}} \frac{\tau_l}{\beta^m} w_l^m \right) \\
& + \sum_{(p,q)\in\mathcal{C}} \left(\int_0^{d_{pq}{}^b} t_{pq}^b(\alpha_{pq}, \rho_{pq}, s)ds + \sum_{m\in\mathcal{M}} \frac{\tau_{pq}}{\beta^m} \right) \\
& + \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} d_{pq}^{nm} t_{pq}^{nm} \\
& + \frac{1}{\theta_1} \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \left[d_{pq}^{nm}(\ln d_{pq}^{nm} - 1) + d_{pq}^{dm}(\ln d_{pq}^{dm} - 1) \right] \\
& - \frac{1}{\theta_2} \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} d_{pq}^{dm}(\ln d_{pq}^{dm} - 1) \\
& + \frac{1}{\theta_2} \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \left[d_{pq}^{cm}(\ln d_{pq}^{cm} - 1) + d_{pq}^{bm}(\ln d_{pq}^{bm} - 1) \right], \qquad (6a)
\end{aligned}$$

subject to

$$d_{pq}^{cm} + d_{pq}^{bm} = d_{pq}^{dm}, \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad (6b)$$

$$d_{pq}^{nm} + d_{pq}^{dm} = d_{pq}^{m}, \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad (6c)$$

$$\sum_{r\in\mathcal{R}_{pq}^m} h_r^m = d_{pq}^{cm}, \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad (6d)$$

$$\sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}^m} \delta_{rl}^m h_r^m = w_l^m, \qquad l \in \mathcal{L}, \quad m \in \mathcal{M}, \qquad (6e)$$

$$\sum_{m\in\mathcal{M}} w_l^m = v_l, \qquad l \in \mathcal{L}, \qquad (6f)$$

$$h_r^m \geq 0, \qquad r \in \mathcal{R}_{pq}^m, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}. \qquad (6g)$$

Positive sign restrictions on the demand variables d_b, d_c, and d^n are unnecessary due to the presence of the logarithms. In fact, in this model all demand variables must take on positive values in the feasible set of problem (6), hence also at an optimal solution. At an optimal solution, the variable d_{pq}^{nm} takes the value of the number of potential travellers in user group m who do not make the trip between p and q, and t_{pq}^{nm} is the corresponding cost, or disutility, of not making the trip.

A sufficient condition for the problem to be convex is that the relation $\theta_2 \geq \theta_1$ is satisfied, which henceforth will be assumed. The existence of an optimal solution is also guaranteed by Weierstrass' Theorem, since the objective function is lower semicontinuous (in fact even continuous) and the (polyhedral) feasible set is closed, nonempty, and bounded. In general one cannot guarantee that the equilibrium route flows (h_r^m) or the user group specific link flows (w_l^m) are unique, while typically the total link flows (v_l) as well as the demands have unique optimal values.

It is an easy exercise to show that the optimality conditions of this problem are equivalent to the conditions describing the nested logit demand model (5) and the statements (1)–(4) of Wardrop equilibrium and the consistency of flows and demands.

3 Definitions of Welfare, Equity and Acceptability Measures

3.1 Welfare Indicator Formulation

The structure of the proposed nested logit model in Fig. 1 locates the decision on the travel mode on the second level and the decision on to travel or not on the first level of the nested logit. This means a traveller will decide first whether he/she will travel, then if he/she decides to travel that traveller will decide upon the mode choice, while the decision on route choice follows Wardrop's user equilibrium (for the car demand given by the mode choice model). Thus, the measurement of utility of the travellers should be made at the highest level of the decision (i.e. decision to travel or not to travel). That is to say, the user equilibrium model is not part of the random utility specification (this would give a different model if it were), but rather is considered as a mapping from the car OD demands to car OD travel costs at equilibrium. Thus, the car OD costs referred to below are equilibrium car OD costs.

In this case, the dispersion parameters for the first and second level are defined as θ_1 and θ_2 (if $\theta_1 = \theta_2$, then the nested model collapses to MNL). At the second level of the decision (mode choice decision), the expected minimum disutility (or satisfaction function) can be defined as

$$S_{pq}^{m,travel} = \frac{1}{\theta_2} \ln \left(\exp \left(\theta_2 [\pi_{pq}^{m,e} - \pi_{pq}^{m,c}] \right) + \exp \left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,b}] \right) \right).$$

This satisfaction function will then be used as the aggregated disutility for the choice of "travel" at the first level of the nested model. We can then define the satisfaction function at the first level as

$$\begin{aligned} S_{pq}^{m} &= \frac{1}{\theta_1} \ln[1 + \exp(\theta_1 s_{pq}^{m,travel})] \\ &= \frac{1}{\theta_1} \ln \left\{ 1 + \exp \left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,c}] \right) + \exp \left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,b}] \right) \right\}. \end{aligned}$$

In the case that $\theta_1 = \theta_2$, we can derive

$$S_{pq}^{m} = \frac{1}{\theta} \ln \left\{ 1 + \exp\left(\ln \left[\exp\left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,c}] \right) + \exp\left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,b}] \right) \right] \right) \right\}$$
$$= \frac{1}{\theta} \ln \left\{ 1 + \exp\left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,c}] \right) + \exp\left(\theta [\pi_{pq}^{m,e} - \pi_{pq}^{m,b}] \right) \right\},$$

which is exactly the satisfaction function for the MNL case.

Net Economic Welfare

The evaluation of the economic welfare involves two components, the consumer surplus and operator surplus:

$$\psi = \text{consumer surplus} + \text{operator surplus}.$$

In general, the consumer surplus can be defined as the benefit (utility) of accessing the destination of the trip subtracted by the generalized cost (dis-utility) of travel. The utility term for travelling is normally a constant. Thus, the part which changes responding to transport policy is the dis-utility of travel. For simplicity, we measure the consumer surplus by the aggregated dis-utility of travel (i.e. satisfaction function). Note that this simplification will not change the result of the analysis in later parts of the chapter. The consumer surplus for the whole network can be defined as

$$S = \sum_{pq} \sum_{m} d_{pq}^{m} S_{pq}^{m},$$

where the operator surplus can be defined as the net financial benefit of the scheme: (scheme revenue − scheme cost). Thus, the welfare measure can be defined as

$$\psi = \sum_{pq} \sum_{m} d_{pq}^{m} S_{pq}^{m} + \text{scheme revenue} - \text{scheme cost}.$$

In the case of MNL, we can use the formulation of the satisfaction function following Eq. (4), and we can use Eq. (6) for the case of nested logit model.

3.2 Inequality Indicator Formulation

There are many ways of measuring inequality, though some candidate measures have undesirable properties: for example, the variance provides a simple measure of the spread of a utility distribution, but it is not independent of the utility scale. This is an undesirable property of an inequality measure, which should not depend on the units of utility adopted. There are five key axioms that are common requirements for an inequality measure.

The Pigou–Dalton Transfer Principle (Pigou [26], Dalton [7])
The inequality measure must fall (or at least not rise) in response to a mean-preserving spread of utility. The transfer of utility from a poorer person to a richer person should result in a fall in the inequality measure, and conversely the transfer of utility from a richer person to a poorer one should result in a rise [2, 3, 5, 27].

Scale Independence
The inequality measure must not depend on the units of utility. Multiplying all utilities by a constant results in the same measure of inequality.

Principle of Population (Dalton [7])
Merging two identical populations (distributions) should not alter the inequality measure.

Anonymity
The inequality measure must be independent of any characteristics of individuals other than their utilities. So the inequality measure is invariant under permutations of the "individuals".

Decomposability
If inequality is seen to rise in each subgroup of the population, then it should rise for the population as a whole.

The Generalized Entropy class of equality measures are easily decomposed into intuitively appealing components of within-group and between-group inequalities. Other measures such as the Atkinson set of equality measures can be similarly decomposed, but the sum of the components' equalities is not the total equality. The Gini coefficient is only decomposable if the partitions are non-overlapping, that is, if the subgroups of the population do not overlap in the vector of utilities (which will not usually be the case). Cowell [6] shows that any equality measure satisfying all of the axioms listed above is a member of the Generalized Entropy class.

3.2.1 The Generalised Entropy Class of Inequality Measures

The generalized entropy class of inequality measures are defined by the formula

$$GE(\alpha) = \frac{1}{\alpha^2 - \alpha}\left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i}{Y^A}\right)\right],$$

where N is the total number of population, Y_i is the utility for individual i, and Y^A is the average utility across the population. The value of GE ranges from 0 (equality) to ∞ (maximum inequality). The parameter α can take any real value and determines the weight placed on inequalities in different parts of the distribution. Common values are:

$\alpha = 0$, which weights differences in the lower tail:

$$GE(0) = \frac{-1}{N} \sum_{i=1}^{N} \log\left(\frac{Y_i}{Y^A}\right);$$

$\alpha = 1$, which weights equally across the distribution (the Theil measure):

$$T = GE(1) = \frac{1}{N} \sum_{i=1}^{N} \frac{Y_i}{Y^A} \log\left(\frac{Y_i}{Y^A}\right);$$

$\alpha = 2$, which gives more weight to the inequality in the upper tail of the distribution (this is, half the squared coefficient of variation)

$$CV = \sqrt{2GE(2)} = \frac{1}{Y^A}\left[\frac{1}{N}\sum_{i=1}^{N}(Y_i - Y^A)^2\right]^{1/2}.$$

3.2.2 Theil's Entropy Measure

Theil's entropy measure belongs to the GE class of inequality measures; it is the difference between the entropy for the actual distribution (of income, or any other values) and the entropy measure of the equal distribution [28]. The basic formula of the Theil's entropy is as follows:

$$T = \frac{1}{N} \sum_{i=1}^{N} \frac{Y_i}{Y^A} \log\left(\frac{Y_i}{Y^A}\right).$$

In our lower level model, the consumer surplus for all travellers from the same user class and the same OD pair is the same. Thus, Theil's entropy measure can be redefined as

$$T = \frac{1}{Q} \sum_{pq} \sum_{m} d_{pq}^m \frac{S_{pq}^m}{S_A} \log\left(\frac{S_{pq}^m}{S_A}\right),$$

where $Q = \sum_{pq} d_{pq}$ denotes the total number of potential travellers in the network and S_A is the average satisfaction function value across travellers from different OD movements and user classes, i.e., $S_A = \frac{1}{Q} \sum_{pq} \sum m d_{pq}^m S_{pq}^m$. This is the aggregated Theil entropy index.[1] As discussed earlier, Theil's entropy index can also be used

[1] Note that $\log N$ (where N is the total demand) is an upper limit of the inequality. This is particularly useful for setting a target.

to analyse the inequality within a group and between groups of individuals. In our case, we define the group by OD movement. Thus, two kinds of inequalities can be measured. The first is the inequality caused by the geography (different OD movement), and the second is the inequality within the OD movement caused by different user classes.

$$T = \left(\frac{1}{Q} \sum_{pq} d_{pq} T_{pq} \frac{S^A_{pq}}{S^A} \right) + \left(\frac{1}{Q} \sum_{pq} d_{pq} T_{pq} \frac{S^A_{pq}}{S^A} \log \left(\frac{S^A_{pq}}{S^A} \right) \right), \tag{7}$$

where $d_{pq} = \sum_m d^m_{pq}$, S^A_{pq} is the mean of consumer surplus within OD pair (p,q) and T_{pq} is Theil's entropy index calculated within the group (from different user classes within each OD pair):

$$T_{pq} = \frac{1}{d_{pq}} \sum_m d^m_{pq} \frac{S^m_{pq}}{\bar{S}} \log \left(\frac{S^m_{pq}}{S^A} \right).$$

The first term on the right-hand side of (7) represents the within-group inequality (the inequality amongst different user classes from the same OD movement) and the second term represents the between-group inequality (the inequality between different OD movements).

3.3 Acceptability Indicator Formulation

Acceptability of a policy can be viewed as an outcome of a democratic process in which the population of a society has a chance to express his/her opinion (to support, be against, or be neutral) on the proposed policy. A common and formal approach to express an opinion is to vote for or against the policy. In the case with multiple choices of policy options, an individual can also vote for different options or for no option. The voting decision of an individual may be modelled as a discrete-choice decision process.

In the traditional discrete-choice theory, an economic agent is assumed to make a choice that offers him/her the highest perceived utility. Often, the utility included in the decision process involves only the individual's utility:

$$U^i_j = V^i_j + \varepsilon^i_j,$$

where U^i_j denotes the utility of policy i offered to individual j, V^i_j is the deterministic utility of policy i offered to individual j, and ε^i_j is the error term. With this utility function, one could derive the probability of policy i to be chosen by individual j under some assumptions about ε^i_j.

However, in the context of acceptability an individual may not consider only the effect of the policy on himself but also the benefit to a wider society (social perspective) [16]. The choice made by a decision maker will have an impact on the others living in the same society [22]. This is indeed an important characteristic of a public policy. Thus, as a member of a society an indirect utility for an individual related to the utility of the whole group may be included into the decision process of an individual [17]. This indirect utility may be the product of either moral or social norm effect. For instance, Mrs. A may vote for a road pricing policy despite being worse-off, since she is aware of the public perception toward this policy (and this is an indirect result from the benefit to the overall society).

With this concept, we can redefine the utility of a transport policy i for individual j as

$$U_j^i = (V_j^i + \sum_g w_g^i \tilde{V}_g^i) + \varepsilon_g^i,$$

where the second term represents the total group utility. For individual j, he or she may consider the utility of option i for a number of groups g. $\tilde{V}_g^i$ denotes the utility of option i offered to group j. w_g^i is the relative weight for the utility offered to group g. These weights represent the relative importance of the change of utility for each group to the decision of individual j. With this modified utility function, we can formulate the probability of an option i to be chosen (or to be accepted) following the standard random utility model. Mueller [23] also suggested a similar framework for voting behaviour but his model interprets the social consideration as the weighted sum of self-individual utility and other individuals' utilities. In our case, we explicitly define a bounded set of population considered as his or her society for each (individual) voter.

With the modified utility function, one needs to define a potential set of groups affecting the decision of different types of individuals. Many social research questions arise with this group definition. For instance, does an individual consider himself/herself as being a member of a group by his or her socio-economics or geography? What are the relative weights for the aggregation of the group effects? The answers to these questions are beyond the scope of the current chapter. Nevertheless, the framework proposed here opens up a number of possibilities in studying the issue of public acceptability both from theoretical and empirical points of view.

In this chapter, we propose to define two types of groups associated with an individual: income group and zoning movement group. In addition, we assume that the error term follows the Gumbel distribution, giving us the logit model for assessing the acceptability of a policy. Following the notation in the previous section, we can define the acceptability level of a policy j as

$$\xi_j = \frac{1}{Q} \sum_{pq} \sum_m d_{pq}^m \frac{\exp\left(\theta V_{pq}^{m,j}\right)}{\sum_{\forall j' \in \Theta} \exp\left(\theta V_{pq}^{m,j'}\right)}, \tag{8}$$

where j denotes a transport policy option in the set of all possible proposed policy options Θ (including do-nothing), $V_{pq}^{m,j}$ denotes the deterministic utility of policy j to users travelling between OD pair pq with class m. Note that in this chapter, we consider the disutility instead of utility offered by different policy options (following the construction of the model presented earlier). As defined earlier, $V_{pq}^{m,j}$ can be defined as a weighted combination of the individual disutility and the group (under his/her consideration) disutility:

$$V_{pq}^{m,j} = d_{pq}^{m} S_{pq}^{m,j} + \sum_{\forall k \neq m} w_{k}^{pq,m} d_{pq}^{k} S_{pq}^{k,j} + \sum_{\forall rs \neq pq} w_{rs}^{pq,m} d_{rs}^{m} S_{rs}^{m,j} \tag{9}$$

where $w_{k}^{pq,m}$ and $w_{rs}^{pq,m}$ are the relative weights given by the travellers from user class m and OD pair pq to the indirect disutilities of travellers from user class k travelling between the same OD pair and to the indirect disutilities of travellers between OD pair rs from the same user class, respectively, in the voting model. The first term of (9) is the direct disutility for user class m travelling between OD pair pq. The second term is the indirect disutility (geographical group utility) from the utilities of the whole users from the same origin–destination. The last term is the indirect disutility (income group utility) from the utilities of the whole users in the network with the income class m.

Indeed in applying this acceptability measure to a real case the calibration of the weighting factors is a crucial task. Similar works have been conducted in other disciplines. Hudson and Jones [15] examine Mueller's concept with the public attitudes to tax and public expenditure. They also conducted a survey to calibrate the weighting factors for individual and social utility.

In applying this acceptability index with the acceptability of a transport policy, we should focus on whether the population will accept a proposed policy or not. Therefore, the relative individual and group disutility adopted in formulation (9) should simply compare the preference between the do-something scenario and the do-nothing scenario. Thus Eq. (8) can be reduced to

$$\xi_1 = \frac{1}{Q} \sum_{pq} \sum_{m} d_{pq}^{m} \frac{\exp\left(-\theta V_{pq}^{m,1}\right)}{\exp\left(-\theta V_{pq}^{m,1}\right) + \exp\left(-\theta V_{pq}^{m,0}\right)}, \tag{10}$$

where we denote the do-something scenario by the superscript "1" and the do-nothing scenario by the superscript "0", and

$$V_{pq}^{m,1} = V_{pq}^{m,do-something} = d_{pq}^{m} S_{pq}^{m,1} + \sum_{\forall k \neq m} w_{k}^{pq,m} d_{pq}^{k} S_{pq}^{k,1} + \sum_{\forall rs \neq pq} w_{rs}^{pq,m} d_{rs}^{m} S_{rs}^{m,1}$$

$$V_{pq}^{m,0} = V_{pq}^{m,do-nothing} = d_{pq}^{m} S_{pq}^{m,0} + \sum_{\forall k \neq m} w_{k}^{pq,m} d_{pq}^{k} S_{pq}^{k,0} + \sum_{\forall rs \neq pq} w_{rs}^{pq,m} d_{rs}^{m} S_{rs}^{m,0}.$$

The indicator proposed in (10) basically measures the proportion of the population who accepts the proposed transport policy. The application of this measure will be illustrated later in Sect. 6.

4 Sensitivity Analysis of Transport Policy Indicators

The optimization model (6) includes several parameters $\gamma = (\rho_l, \tau_l, \alpha_{pq}, \rho_{pq}, \tau_{pq})$ in the objective function, while the feasible set does not involve any; this implies that we have access to a powerful sensitivity analysis tool presented in [24, 25], and further adapted to the traffic equilibrium problem in Josefsson and Patriksson [24].

We will utilize the Lagrangian formulation and state the equilibrium model as a mixed complementarity system; in fact, problem (6) is reduced to a model which "nearly" is a system of nonlinear equations, except for the complementarity system stemming from the Wardrop conditions on the route flows. The problem to be analysed is the following parameterized variational inequality:

$$-f(\gamma, x) \in N_C(x),$$

where γ is a vector of parameters, x is the vector of variables, C is a fixed polyhedral set, and N_C denotes the normal cone to the set C.

In the sensitivity analysis of the equilibrium solution we consider adjusting one or several parameter values along some direction $(\rho_l', \tau_l', \alpha_{pq}', \rho_{pq}', \tau_{pq}')$ and ask what the resulting perturbation (that is, rate of change) of the equilibrium solution is.

4.1 Application to the Present Problem

We first set up our model according to the above framework, let

$$x = \begin{pmatrix} (h_r^m) \\ (d_{pq}^{bm}) \\ (d_{pq}^{cm}) \\ (d_{pq}^{nm}) \\ (d_{pq}^{dm}) \\ (\mu_{pq}^m) \\ (\eta_{pq}^m) \\ (\pi_{pq}^m) \\ (w_l^m) \\ (v_l) \end{pmatrix};$$

$$C = \mathbf{R}_+^{|\mathcal{R}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times$$
$$\mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{C}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{L}|\times|\mathcal{M}|} \times \mathbf{R}^{|\mathcal{L}|};$$

and

$$\Phi(\gamma,x) = \begin{pmatrix} \left(\sum_{l\in\mathcal{L}} \delta_{rl}^m [t_l^c(\rho_l, v_l) + \frac{\tau_l}{\beta^m}] - \pi_{pq}^m\right)_{r\in\mathcal{R}_{pq}^m, m\in\mathcal{M}} \\ \left(t_{pq}^b(\alpha_{pq},\rho_{pq},d_{pq}^b) + \frac{1}{\theta_2}\ln d_{pq}^{bm} - \mu_{pq}^m\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(\frac{1}{\theta_2}\ln d_{pq}^{cm} - \mu_{pq}^m + \pi_{pq}^m\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(t_{pq}^{nm} + \frac{1}{\theta_1}\ln d_{pq}^{nm} - \eta_{pq}^m\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(\frac{1}{\theta_1}\ln d_{pq}^{dm} - \frac{1}{\theta_2}\ln d_{pq}^{dm} + \mu_{pq}^m - \eta_{pq}^m\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(d_{pq}^{cm} + d_{pq}^{bm} - d_{pq}^{dm}\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(d_{pq}^{nm} + d_{pq}^{dm} - d_{pq}^{m}\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(\sum_{r\in\mathcal{R}_{pq}^m} h_r^m - d_{pq}^{cm}\right)_{(p,q)\in\mathcal{C},m\in\mathcal{M}} \\ \left(\sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}^m} \delta_{rl}^m h_r^m - w_l^m\right)_{l\in\mathcal{L},m\in\mathcal{M}} \\ \left(\sum_{m\in\mathcal{M}} w_l^m - v_l\right)_{l\in\mathcal{L}} \end{pmatrix}.$$

The above problem represents our equilibrium problem: the top row of Φ is simply the Lagrangian cost of the route flows, and the corresponding part of the set C, that is, $\mathbf{R}_+^{|\mathcal{R}|\times|\mathcal{M}|}$, represents the non-negativity restriction on the route flow variables (here, for simplicity, in the notation $|\mathcal{R}|\times|\mathcal{M}|$ assuming that all the users have access to the same set of routes). Hence, the top row is nothing but the Wardrop conditions. The remaining conditions state the nested logit demand structure and the conservation and consistency conditions on the network flows.

Suppose now that the parameter vector γ^* is given, and that for this value the corresponding solution is x^*. Suppose also that the link travel time function t is such that at the given vector γ^* it is strictly monotone in v and moreover has a partial Jacobian with respect to v which is positive definite at v^*. Further, we must also assume that $\theta_2 > \theta_1$, in order to ensure the uniqueness of the demand and its perturbation. These conditions will ensure that the equilibrium link flows and demands are unique for each value of γ^*, and that it is directionally differentiable in any direction of change γ'; also, this value is found as the unique link flow and demand perturbations solving the sensitivity problem.

Theory also states that the equilibrium solution is differentiable if, and only if, every route that has a zero flow in the original equilibrium solution has a zero flow after every small enough perturbation in the cost functions. (This is what the linearity of the directional derivative amounts to in our present context; strict complementarity is a sufficient but *not* a necessary condition for this property to hold.) It is then possible to produce the entire gradient of the equilibrium solution, which is then assembled from the directional derivatives at each unit coordinate direction γ'.

It remains to investigate the appearance of this problem in our context:

- The sign conditions on the route flow perturbation variables h' are not "≥ 0" as in the original model. Here, they are as follows: if $(h_r^m)^* > 0$ then $(h_r^m)'$ is a free variable; if $(h_r^m)^* = 0$ but this route is a shortest route, then $(h_r^m)' \geq 0$ should hold; if $(h_r^m)^* = 0$ and this route is not a shortest route, then $(h_r^m)' = 0$ should hold.

  Notice the important fact that the sensitivity results are placed at the aggregated level and is not affected by the choice of a particular equilibrium route flow: the choice of h^* is completely arbitrary.
- The travel cost function $v_l \mapsto t_l(\rho_l^*, v_l)$ is changed into the following:

$$\frac{\partial t_l}{\partial \rho_l}(\rho_l^*, v_l^*)\rho_l' + \frac{\partial t_l}{\partial v_l}(\rho_l^*, v_l^*)v_l' + \frac{\tau_l'}{\beta^m}.$$

  In other words, the perturbed cost is the sum of a fixed cost given by the dependence on ρ_l in the cost formula, and a term which is given by the link-flow derivative of the link cost. The optimization formulation of the sensitivity problem will hence have a quadratic objective with respect to the link flow variables, and since, by assumption, $\frac{\partial t_l}{\partial v_l}(\rho_l^*, v_l^*) > 0$ holds for all $l \in \mathcal{L}$ it is also strictly convex.
- The demand functions are similarly changed into the following:

$$\frac{\partial t_{pq}^b}{\partial \rho_{pq}}(\alpha_{pq}^*, \rho_{pq}^*, (d_{pq}^b)^*)\alpha_{pq}' + \frac{\partial t_{pq}^b}{\partial \alpha_{pq}}(\alpha_{pq}^*, \rho_{pq}^*, (d_{pq}^b)^*)\alpha_{pq}' + \frac{\tau_{pq}'}{\beta^m} + \frac{1}{\theta_2}\frac{1}{(d_{pq}^{bm})^*}(d_{pq}^{bm})';$$

$$\frac{1}{\theta_2}\frac{1}{(d_{pq}^{cm})^*}(d_{pq}^{cm})';$$

$$\frac{1}{\theta_1}\frac{1}{(d_{pq}^{nm})^*}(d_{pq}^{nm})';$$

$$\left(\frac{1}{\theta_1} - \frac{1}{\theta_2}\right)\frac{1}{(d_{pq}^{dm})^*}(d_{pq}^{dm})';$$

  for d^b, d^c, d^n, and d^d travel demands, respectively. Notice that also for the demand variables the sum of integrals of the above cost functions is strictly convex, due to the property that every demand variable is positive at equilibrium and $\theta_2 > \theta_1$ holds.

Let, for short, $x' = ((h')^{\mathrm{T}}, (w')^{\mathrm{T}}, (v')^{\mathrm{T}}, (d^{b'})^{\mathrm{T}}, (d^{c'})^{\mathrm{T}}, (d^{n'})^{\mathrm{T}})^{\mathrm{T}}$. The optimization problem for the sensitivity has the following form:

$$\begin{aligned}
\text{minimize}\, \phi'(x') := & \sum_{l\in\mathcal{L}} \left[\frac{1}{2} \frac{\partial t_l(\rho_l^*, v_l^*)}{\partial v_l} (v_l')^2 + \frac{\partial t_l(\rho_l^*, v_l^*)}{\partial \rho_l} \rho_l' v_l' + \sum_{m\in\mathcal{M}} \frac{\tau_l'}{\beta^m} (w_l^m)' \right] \\
& + \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \left[\frac{\partial t_{pq}^b(\alpha_{pq}^*), \rho_{pq}^*, (d_{pq}^b)^*)}{\partial \alpha_{pq}} \alpha_{pq}' (d_{pq}^{bm})' \right. \\
& \left. + \frac{\partial t_{pq}^b(\alpha_{pq}^*), \rho_{pq}^*, (d_{pq}^b)^*)}{\partial \rho_{pq}} \rho_{pq}' (d_{pq}^{bm})' + \frac{\tau_{pq}'}{\beta^m} (d_{pq}^{bm})' \right] \\
& + \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \frac{1}{2\theta_2} \frac{1}{(d_{pq}^{bm})^*} \left((d_{pq}^{bm})' \right)^2 \\
& + \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \frac{1}{2\theta_2} \frac{1}{(d_{pq}^{cm})^*} \left((d_{pq}^{cm})' \right)^2 \\
& + \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \frac{1}{2\theta_1} \frac{1}{(d_{pq}^{nm})^*} \left((d_{pq}^{nm})' \right)^2 \\
& + \sum_{(p,q)\in\mathcal{C}} \sum_{m\in\mathcal{M}} \frac{1}{2} \left(\frac{1}{\theta_1} - \frac{1}{\theta_2} \right) \frac{1}{(d_{pq}^{dm})^*} \left((d_{pq}^{dm})' \right)^2, \qquad \text{(11a)}
\end{aligned}$$

subject to

$$(d_{pq}^{cm})' + (d_{pq}^{bm})' = (d_{pq}^{dm})', \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad \text{(11b)}$$

$$(d_{pq}^{nm})' + (d_{pq}^{dm})' = 0, \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad \text{(11c)}$$

$$\sum_{r\in\mathcal{R}_{pq}} (h_r^m)' = (d_{pq}^{cm})', \qquad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad \text{(11d)}$$

$$\sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{rl}^m (h_r^m)' = (w_l^m)', \qquad l \in \mathcal{L}, \quad m \in \mathcal{M}, \qquad \text{(11e)}$$

$$\sum_{m\in\mathcal{M}} (w_l^m)' = (v_l)', \qquad l \in \mathcal{L}. \qquad \text{(11f)}$$

$$(h_r^m)' \text{ free}, \qquad r \in (\mathcal{R}_{pq}^m)^1, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad \text{(11g)}$$

$$(h_r^m)' \geq 0, \qquad r \in (\mathcal{R}_{pq}^m)^2, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad \text{(11h)}$$

$$(h_r^m)' = 0, \qquad r \in (\mathcal{R}_{pq}^m)^3, \quad (p,q) \in \mathcal{C}, \quad m \in \mathcal{M}, \qquad \text{(11i)}$$

where $(\mathcal{R}_{pq}^m)^1$ is the subset of the routes where $(h_r^m)^* > 0$, the set $(\mathcal{R}_{pq}^m)^2$ consists of the routes where $(h_r^m)^* = 0$ but the route is an equilibrium route (that is, has a minimal cost) and $(\mathcal{R}_{pq}^m)^3$ is the set of routes for which $(h_r^m)^* = 0$ and the route is not an equilibrium route.

Solving problem (11) provides the demand and link flow perturbation from the marginal change in the design parameters.

Some of the aggregate sensitivity measures can be found as dual variables to the constraints; for example, the OD pair cost perturbations by travel mode can be found from (11d). The OD pair cost perturbations for the other modes cannot be found directly from the solution to the optimization model (11). However, these costs can be found also by introducing additional definitional constraints in the model or by computing them based on the demand and flow perturbations.

The OD pair cost perturbations for the car mode can directly be computed by

$$(\pi_{pq}^{cm})' = \min_{r \in \mathcal{R}_{pq}^m} \sum_{l \in \mathcal{L}} \delta_{rl}^m \left(\frac{\partial t_l(\rho_l^*, v_l^*)}{\partial v_l} v_l' + \frac{\partial t_l(\rho^*, v_l^*)}{\partial \rho_l} \rho_l' + \frac{\tau_l'}{\beta_m} \right),$$

where $\mathcal{R}_{pq}$ includes the two route sets $(\mathcal{R}_{pq}^m)^1$ and $(\mathcal{R}_{pq}^m)^2$. The OD pair cost perturbations for the bus mode is given by

$$(\pi_{pq}^{bm})' = \frac{\partial t_{pq}^{bm}((d_{pq}^{bm})^*, \alpha_{pq}^*)}{\partial \alpha_{pq}} \alpha_{pq}' + + \frac{\tau_{pq}'}{\beta_m}.$$

5 Numerical Experiments

Numerical tests have been made on a Norrköping, Sweden, case network. The case includes a network model of the street network for the car trips and a tram network. Data for the spatial description of the network, the travel demands, and travel elasticity parameters has been based on data supplied by the Norrköping municipality and has been extracted from their VIPS traffic model. The spatial description of the street network consists of 1,251 links, 500 nodes, and 7,660 travel relations modelling the potential trips during the morning peak hour. The travel demand used is an estimated future demand for the year 2035. One user group is modelled. The two tram lines in Norrköping are modelled. A generalized travel time (in minutes) for using the tram is given for each travel relation. The generalized travel times include walking times, waiting times, and in-vehicle times. A walking time estimate is computed by multiplying the distance from each origin node to the nearest tram line by an average walking speed of 5 km/h. The same is made for each destination node. The waiting time is estimated to 60 over the tram frequency for the tram line divided by two. The in-vehicle tram time is computed based on the tram travel distance and an average tram speed. The logit model elasticity is calibrated such that in the equilibrium solution 5 % of the assigned travel demand is assigned to the tram network.

Fig. 2 Norrköping equilibrium flows

5.1 Equilibrium Model Results

The traffic equilibrium problem is solved using a procedure based on the partial linearization scheme of Evans [9]. A computer code has been implemented in Matlab. The equilibrium problem is solved, and in each iteration re-optimized, using a route based procedure (see Larsson and Patriksson [21]). The car network model is depicted in Fig. 2. The link widths in the figure are proportional to the car flow on the links.

5.2 Sensitivity Analysis Results

We provide three example uses of the sensitivity computations.

In the first one we have introduced a unit toll on the bridges in the central parts of Norrköping and computed the directional derivatives on link flows and travel demands with respect to these tolls. The unit toll corresponds here to an increase in the generalized travel time on each link by 1 min. The placements of the four tolls are marked in Fig. 3 by the thick black lines.

Results from this bridge toll experiment can be seen in Fig. 4, where dark (red) links indicate a decrease in car flows and lighter (green) links indicate an increase in

Fig. 3 Equilibrium flows for the central parts of Norrköping

Fig. 4 Norrköping equilibrium car flow changes with the bridge tolls

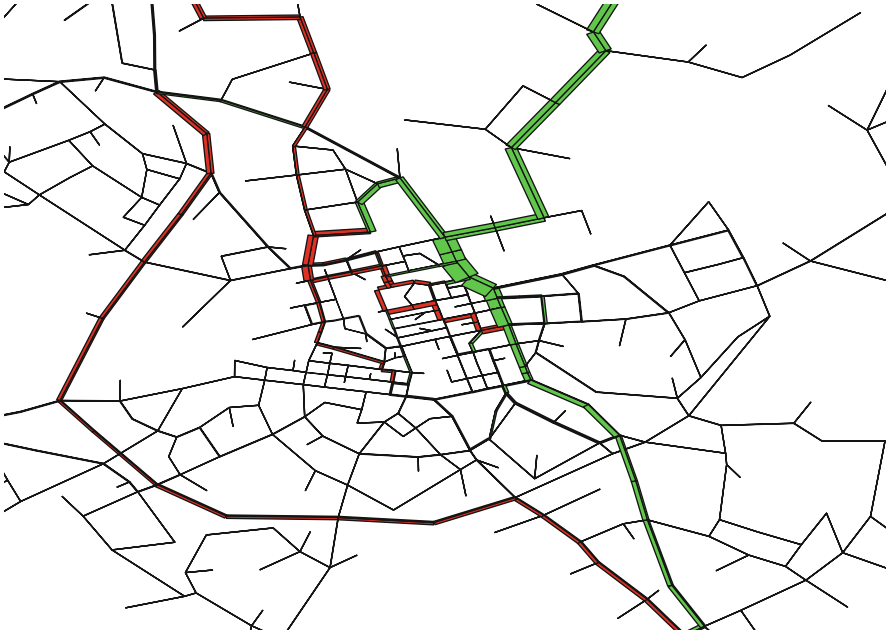


Fig. 5 Norrköping equilibrium car flow changes with increased link capacity on the bridge Hamnbron

car flows. The link widths are proportional to the level of the flow change. (Note that the scale is different in the sensitivity result figures compared to that in Fig. 3.) From the sensitivity information we have also observed that the total number of travellers in the network will decrease but the number of tram trips will increase, as a result from these tolls.

In the second example we have increased the link capacity on the bridge Hamnbron. The bridge is marked in Fig. 2 as the right-most black line. The resulting changes in car traffic flow are shown in Fig. 5. Links with lighter colour (green) have got an increase in car flows, and links with darker colour (red) have got a decrease in car flows. From the figure it is noted that routes passing the bridge where the capacity is increased have got an increase in flow.

The third example is constructed by increasing the tram fare. In Fig. 6 the change in car traffic flows is shown, as resulting from a marginal change in the tram fare. Almost all links have got an increased car flow, as a result of an increase of the car demand, and a decrease in tram trips.

Ultimately, these forecasts may be used to examine changes in the evaluation measures per unit increase/decrease in the policy variables. Below, these results are shown, first, in Fig. 7, for the case of uniform capacity increases to the two main bridges in Norrköping, and second, in Fig. 8, for the case of uniform tolls (expressed in units of an equivalent travel time penalty) applied to these bridges. Note that the only reason to choose uniform changes is for ease of illustration on

Fig. 6 Norrköping equilibrium car flow changes from increased tram fares

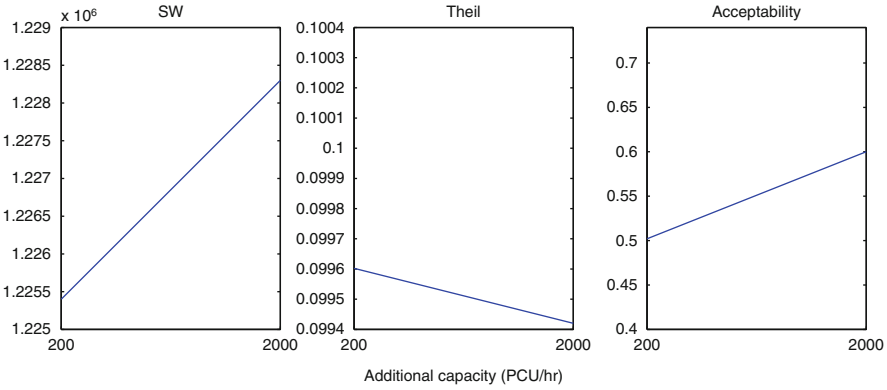


Fig. 7 Bridge capacity sensitivity analysis

a two-dimensional plot—the analysis is otherwise quite general. For the toll case, the plots illustrate that while overall social welfare (SW) increases over the range of tolls considered, this comes at the price of an increase in inequality, and that public acceptability will decline with higher tolls. The increase in capacity is, however, attractive from the viewpoint of all three measures, though in practice there are likely to be other targets (e.g. a reduction of emissions from car traffic) that are likely to weigh against such considerations. In time, given experience with the scale and

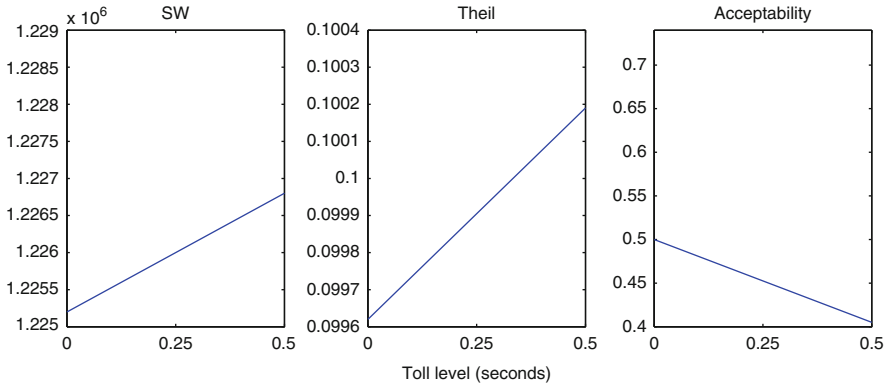


Fig. 8 Bridge toll sensitivity analysis

interpretation of such results, the intention is that planners will be able to trade off the aggregate efficiency with the equitable distribution of benefits and their public acceptance.

6 Conclusions

We have developed a multi-modal traffic equilibrium model with which one can predict changes in traffic flows, congestion effects, and travel demands and costs, as parameters in costs and demands change. With this computational model one can analyse the effect of socio-economic policies on a given traffic system, with the end goal of proposing an optimal policy. To that end, we have also developed an algorithm for computing the traffic equilibrium flows and demands and their sensitivities to changes in parameter values, such as link tolls and bus fares, and established for a realistic test case that it is also efficient in producing reliable solutions.

Planned future research topics include the construction of a hierarchical (that is, bi-level) optimization model through which one can automatically select the best policy, according to the specification of socio-economic measures of equity, and the testing of the model on realistic network data.

Acknowledgements This research is funded by Volvo Research Foundation, Volvo Educational Foundation, and Dr Pehr G. Gyllenhammar Research Foundation.

References

1. Abrahamsson, T., Lundqvist, L.: Formulation and estimation of combined network equilibrium models with applications to Stockholm. Transp. Sci. **33**, 80–100 (1999)
2. Atkinson, A.B.: On the measurement of inequality. J. Econ. Theory **2**, 244–263 (1970)
3. Atkinson, A.B.: The Economics of Inequality. Clarendon Press, Oxford (1983)
4. Beckmann, M., McGuire, C.B., Winsten, C.B.: Studied in the Economics of Transportation. Yale University Press, New Haven (1956)
5. Cowell, F.A.: Measures of distributional change: an axiomatic approach. Rev. Econ. Stud. **52**, 135–151 (1985)
6. Cowell, F.A.: Measuring Inequality. Harvester Wheatsheaf, Hemel Hempstead (1995)
7. Dalton, H.: The measurement of the inequality of incomes. Econ. J. **30**, 348–361 (1920)
8. de Dios Ortúzar, J., Willumsen, L.G.: Modelling Transport, 2nd edn. Wiley, Chichester (1996)
9. Evans, S.P.: Derivation and analysis of some models for combining trip distribution and assignment. Transp. Res. **10**, 37–57 (1976)
10. Fernandez, E., De Cea, J., Florian, M., Cabrera, E.: Network equilibrium models with combined modes. Transp. Sci. **28**, 182–192 (1994)
11. Florian, M.: A traffic equilibrium model of travel by car and public transit modes. Transp. Sci. **2**, 166–179 (1977)
12. Florian, M., Nguyen, S.: A combined trip distribution modal split and trip assignment model. Transp. Res. **12**, 241–246 (1978)
13. Florian, M., Nguyen, S., Ferland, J.: On the combined distribution—assignment of traffic. Transp. Sci. **9**, 43–53 (1975); Erratum Transp. Sci. **9**, 173 (1975)
14. Florian, M., Wu, J.-H., He, S.: A multi-class multi-mode variable demand network equilibrium model with hierarchical logot structures. In: Gendreau, M., Marcotte, P. (eds.) Transportation and Network Analysis: Current Trends. Applied Optimization, vol. 63, pp. 237–243. Kluwer Academic Publishers, Dordrecht (2002)
15. Hudson, J., Jones, P.: The importance of the "ethical voter": an estimate of "altruism". Eur. J. Polit. Econ. **10**, 499–509 (1994)
16. Jaensirisak, S., May, A.D., Wardman, M.: Acceptability of road user charging: the influence of selfish and social perspectives. In: Schade, J., Schlag, B. (eds.) Acceptability of Transport Pricing Strategies, pp. 203–218. Elsevier, Oxford (2003)
17. Jaensirisak, S., Wardman, M., May, A.D.: Explaining variations in public acceptability of road pricing schemes. J. Transp. Econ. Policy **39**(Part 2), 127–153 (2005)
18. Josefsson, M., Patriksson, M.: Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design. Transp. Res. B Methodol. **41**(1), 4–31 (2007)
19. Kolm, S.C.: Public economics. In: International Economic Association Conference on Public Economics, Biarritz, Proceedings, Economie Publique The Optimal Production of Social Justice, pp. 109–177. McMillan, CNRS, Paris (1969)
20. Langmyhr, T.: Managing equity: the case of road pricing. Transp. Policy **4**(1), 25–39 (1997)
21. Larsson, T., Patriksson, M.: Simplicial decomposition with disaggregated representation for the traffic assignment problem. Transp. Sci. **26**, 4–17 (1992)
22. Margolis, H.: Dual utilities and rational choice. In: Mansbridge, J.J. (ed.) Beyond Self-Interest, pp. 239–253. The University of Chicago Press, Chicago, IL (1990)
23. Mueller, D.C.: Rational egoism versus adaptive egoism as fundamental postulate for a descriptive theory of human behaviour. Public Choice **51**, 3–23 (1986)
24. Patriksson, M.: Sensitivity analysis of traffic equilibria. Transp. Sci. **38**, 258–281 (2004)
25. Patriksson, M., Rockafellar, R.T.: Sensitivity analysis of aggregated variational inequality problems, with application to traffic equilibria. Transp. Sci. **37**, 56–68 (2003)
26. Pigou, A.F.: The Economics of Welfare. Macmillan, London (1912)
27. Sen, A.K.: On Economic Inequality. Oxford University Press, London (1973)

28. Theil, H.: Economics and Information Theory. North-Holland, Amsterdam (1967)
29. Wong, K.I., Wong, S.C., Wu, J.-H., Yang, H., Lam, W.H.K.: A combined distribution, hierarchical mode choice, and assignment model with multiple user and mode classes. In: Lee, D.-H., Boyce, D.E. (eds.) Urban and Regional Transportation Modeling: Essays in Honor of David Boyce, pp. 25–42. Edward Elgar, Northhampton (2004)

Calibration in Survey Sampling as an Optimization Problem

Gareth Davies, Jonathan Gillard, and Anatoly Zhigljavsky

Dedicated to Professor Panos Pardalos on the occasion of his 60th birthday

Abstract Calibration is a technique of adjusting sample weights routinely used in sample surveys. In this chapter, we consider calibration as an optimization problem and show that the choice of optimization function has an effect on the calibrated weights. We propose a class of functions that have several desirable properties, which includes satisfying necessary range restrictions for the weights. In this chapter, we explore the effect these new functions have on the calibrated weights.

Keywords Sampling calibration • Weights calibration optimization problem • g-weights

1 Introduction

Calibration of survey samples is one of the key issues in official statistics and analysis of panel data (in particular, in market research). The problem of calibration can be defined informally as follows. Suppose there are some initial weights $d_1, \ldots, d_n$ assigned to n objects of a survey. Suppose further that there are m auxiliary variables and that for these auxiliary variables the sample values are known, either exactly or approximately. The calibration problem seeks to improve the initial weights by finding new weights $w_1, \ldots, w_n$ that incorporate the auxiliary information. In a typical practical problem, the sample size n is rather large (samples of order 10^4 and larger are very common). The number of auxiliary variables m can also be large although it is usually much smaller than n.

Three main reasons are advocated for using calibration in practice (see for example [2]). The first of these is to produce estimates consistent with other sources

G. Davies • J. Gillard • A. Zhigljavsky (✉)
Cardiff School of Mathematics, Cardiff University, Cardiff, UK
e-mail: DaviesGP2@cardiff.ac.uk; GillardJW@cardiff.ac.uk; ZhigljavskyAA@cardiff.ac.uk

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_4

of data. Indeed, when a statistical office publishes the same statistics via two data sources, the validity of the statistics will be questioned if there are contradictions between the sources. The second reason is to reduce the sampling variance of estimates as the inclusion of the additional calibration information can lead to a reduction in the variance of the estimators (see for example [11]). The third argument for calibration is a reduction of the coverage and/or non-response bias (see for example [10]).

In this chapter, we properly formulate the problem of calibration of weights as an optimization problem, study properties of the corresponding optimization problems and give recommendations on how to choose the objective function. We claim that the literature on calibration has ignored this important issue which lead to the recipes which were inefficient or even incorrect.

Notation

We use the following key notation throughout the chapter:

$D = (d_1, \ldots, d_n)'$:	Vector of initial weights,
$W = (w_1, \ldots, w_n)'$:	Vector of calibrated weights,
$G = (g_1, \ldots, g_n)'$:	Vector of the g-weights $g_i = w_i/d_i$,
$L = (l_1, \ldots, l_n)'$:	Vector of lower bounds for the g-weights,
$U = (u_1, \ldots, u_n)'$:	Vector of upper bounds for the g-weights,
$X = (x_{ij})_{i,j=1}^{n,m}$:	Given $n \times m$ matrix,
$A = (a_{ij})_{i,j=1}^{n,m}$:	$n \times m$ matrix with entries $a_{ij} = d_i x_{ij}$,
$T = (t_1, \ldots, t_m)'$:	An arbitrary $m \times 1$ vector,
$\mathbf{1} = (1, 1, \ldots, 1)'$	$n \times 1$ vector of ones,
$\mathbb{G}$	Feasible domain in the calibration problem.

2 Calibration as an Optimization Problem

A vector of initial weights $D = (d_1, \ldots, d_n)'$ is given. The d_i are always assumed to be positive: $d_i > 0$ for all i. Our aim is to calibrate (improve) these initial weights in view of some additional information. The vector of calibrated (improved) weights will be denoted by $W = (w_1, \ldots, w_n)'$.

We are given a matrix $X = (x_{ij})_{i,j=1}^{n,m}$ of realizations of m auxiliary variables. The (i,j)th entry x_{ij} of X denotes the value which the ith member of the sample takes on the jth auxiliary variable. Formally, X is an arbitrary $n \times m$ matrix. Given the vector $T = (t_1, \ldots, t_m)'$, exact (hard) constraints can be written as $X'W = T$, whereas approximate (soft) constraints are $X'W \simeq T$. These constraints, whether exact or approximate, define the additional information we use in the calibration of the weights.

It is sometimes natural to impose a constraint on the sum of the weights. In this chapter, we shall consider the sum of weights constraint $\sum_{i=1}^{n} w_i = \sum_{i=1}^{n} d_i$ or, in vector notation, $\mathbf{1}'W = \mathbf{1}'D$, where $\mathbf{1} = (1,1,\ldots,1)'$. This constraint is motivated in [17]. The condition $\mathbf{1}'W = \mathbf{1}'D$ can be added to the set of the main constraints $X'W = T$ (see, for example [16]). Hence we do not formally distinguish the cases when the condition $\mathbf{1}'W = \mathbf{1}'D$ is required or not.

In most practical cases of survey sampling and panel data analysis, the ratios of the weights w_i and d_i are of prime importance rather than the weights w_i themselves and the so-called g-weights $g_i = w_i/d_i$ are considered. Denote the vector of g-weights by $G = (g_1,\ldots,g_n)'$ and consider this vector as the vector of calibrated weights we are seeking.

Since $d_i > 0$ for all i, the hard constraints $X'W = T$ can be written in the form $A'G = T$, where the matrix $A = (a_{ij})_{i,j=1}^{n,m}$ has elements $a_{ij} = d_i x_{ij}$. Correspondingly, soft constraints $X'W \simeq T$ have the form $A'G \simeq T$.

In addition to either hard or soft constraints, the following constraints on G have to be imposed. First of all, the calibrated weights must be nonnegative, that is $g_i \geq 0$ for all i. Moreover, much of the calibration literature, see for example [4] and [18], recommends imposing stricter constraints on the g-weights of the form $L \leq G \leq U$, where $L = (l_1,\ldots,l_n)'$ and $U = (u_1,\ldots,u_n)'$ are some given $n \times 1$ vectors such that $0 \leq l_i < 1 < u_i \leq \infty$ for all i. That is the g-weights should satisfy $l_i \leq g_i \leq u_i$ for some sets of lower and upper bounds l_i and u_i. If $l_i = 0$ and $u_i = \infty$ for all i, then the constraint $l_i \leq g_i \leq u_i$ coincides with the simple non-negativity constraint $g_i \geq 0$. In the majority of practical problems $l_i = l$ and $u_i = u$ for all i with $0 \leq l < 1 < u \leq \infty$, where the strict inequalities $l > 0$ and $u < \infty$ are very common.

In the process of calibration, the weights W have to stay as close as possible to the initial weights D. Equivalently, the g-weights G have to stay as close as possible to the vector $\mathbf{1}$. To measure the closeness of G and $\mathbf{1}$, we use some function $\Phi(G) = \Phi(g_1,\ldots,g_n)$. This function is required to satisfy the following properties (see [5] for a related discussion): (a) $\Phi(G) \geq 0\ \forall G$, (b) $\Phi(\mathbf{1}) = 0$, (c) $\Phi(G)$ is twice continuously differentiable, and (d) $\Phi(G)$ is strictly convex. The function Φ often has the form

$$\Phi(G) = \Phi(g_1,\ldots,g_n) = \sum_{i=1}^{n} q_i \phi_i(g_i)\,, \tag{1}$$

where $q_1,\ldots,q_n$ are given non-negative numbers; in the majority of applications $q_i = d_i$ for all i. We shall concentrate on this form of Φ; in Sect. 3, we discuss the choice of the functions ϕ_i.

Hard constraints $A'G = T$ enter the definition of the feasible domain of G. Soft constraints $A'G \simeq T$ can either enter the definition of the feasible domain of G in the form $\|A'G - T\| \leq \varepsilon$ for some vector norm $\|\cdot\|$ and some given $\varepsilon > 0$, or can be

put as a penalty $\Psi(A'G,T)$ into the objective function. The properties required for Ψ (as a function of G) are similar to those required for Φ. The most common choice for Ψ is

$$\Psi(A'G,T) = \beta(A'G-T)'C(A'G-T), \tag{2}$$

where C is some user-specified $m \times m$ positive definite (usually, diagonal) matrix and $\beta > 0$ is some constant (see for example [2, equation (2.3)]).

Summarizing, we have the following versions of the calibration problem formulated in terms of the g-weights G.

Hard constraint case:

$$\Phi(G) \to \min_{G\in\mathbb{G}}, \text{ where } \mathbb{G} = \{G : L \le G \le U \text{ and } A'G = T\}. \tag{3}$$

Soft constraint case I:

$$\Phi(G) \to \min_{G\in\mathbb{G}}, \text{ where } \mathbb{G} = \{G : L \le G \le U \text{ and } \Psi(A'G,T) \le 1\}. \tag{4}$$

Soft constraint case II:

$$\Phi(G) + \Psi(A'G,T) \to \min_{G\in\mathbb{G}}, \text{ where } \mathbb{G} = \{G : L \le G \le U\}. \tag{5}$$

In problems (3)–(5), the matrix A and the vectors T, L and U are given, and in the majority of applications the functions Φ and Ψ have the forms (1) and (2) correspondingly.

Optimization problems (3) and (4) may have no solutions, that is the feasible domain $\mathbb{G}$ in these problems may be empty. The case when $\mathbb{G}$ is empty means that the constraints on G are too strong. The feasible domain $\mathbb{G}$ in problem (5) is always non-empty and the optimal solution always exists. In view of the strict convexity of Φ and Ψ as well as the compactness of $\mathbb{G}$, if the optimal solution exists then it is necessarily unique. Optimization problem (4) is considered too difficult by practitioners and hence it is never considered (despite it looking rather natural). We therefore consider problems (3) and (5) only.

3 Choice of Functions ϕ_i in (1)

Here we discuss the choice of the functions ϕ_i in (1). See Sect. 4 for examples of calibrated weights obtained using different forms of functions ϕ_i. By slightly modifying the assumptions of [4], we require the function $\phi_i : (l_i, u_i) \to \mathbb{R}_+$ to satisfy the following properties: (i) $\phi_i(g) \ge 0$ for all $g \in (l_i, u_i)$, (ii) $\phi_i(1) = 0$, (iii) ϕ_i is twice continuously differentiable and strictly convex. The function ϕ_i does not have to be defined outside the open interval (l_i, u_i). If all ϕ_i satisfy conditions (i)–(iii) then the function Φ defined in (1) satisfies conditions (a)–(d) formulated above.

Since these functions are chosen in the same manner for all i, the subscript i will be dropped and the function ϕ_i will be denoted simply by ϕ. Correspondingly, the lower and upper bounds l_i and u_i for the g-weights g_i will be denoted by l and u, respectively.

We will illustrate the shape of several functions ϕ in Figs. 1, 2, and 3. In all these figures, we choose $l = 1/4, u = 4$ and plot all the functions in the interval $(l, u) = (\frac{1}{4}, 4)$, despite some of the functions are defined in a larger region. As our intention in this section is illustrating shapes of the possible calibration functions ϕ

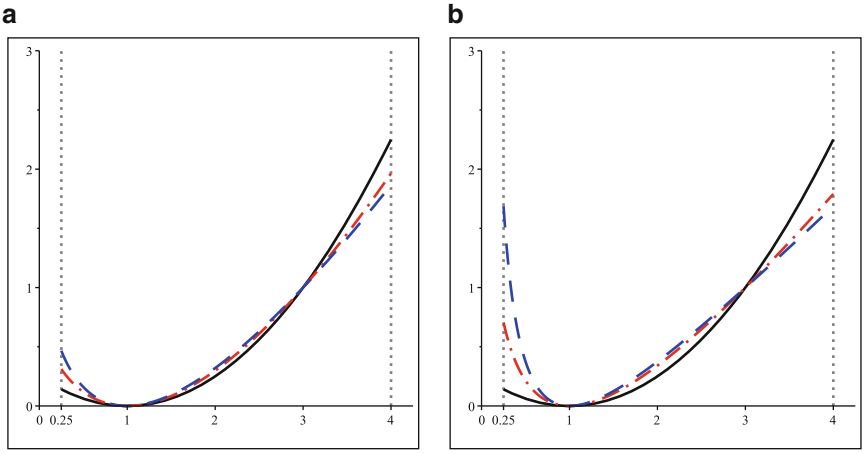


Fig. 1 Classical calibration functions of Type I scaled so that $c_k\phi^{(k)}(3) = 1$, $k = 1, \ldots, 5$. (**a**) $\phi^{(1)}$ (*line*), $\phi^{(2)}$ (*dot-dash*) and $\phi^{(3)}$ (*dash*). (**b**) $\phi^{(1)}$ (*line*), $\phi^{(4)}$ (*dot-dash*) and $\phi^{(5)}$ (*dash*)

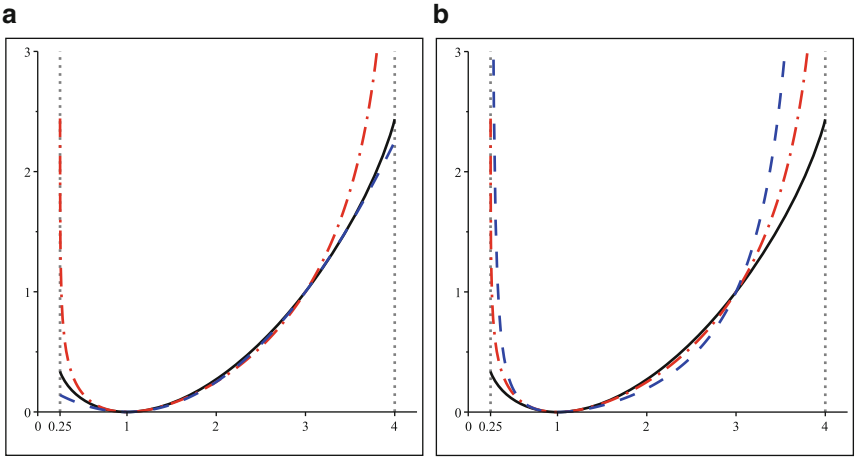


Fig. 2 Functions $\phi^{(1)}$, $\phi^{(6)}$, $\phi^{(7)}$ and $\phi^{(8)}$ scaled so that $c_1\phi^{(1)}(3) = 1$ and $c_k\phi^{(k)}(3; \frac{1}{4}, 4) = 1$, $k = 6, 7$ and $c_{8,1}\phi^{(8)}(3; \frac{1}{4}, 4, 1) = 1$. (**a**) $\phi^{(6)}$ (*line*) and $\phi^{(7)}$ (*dot-dash*) and $\phi^{(1)}$ (*dash*). (**b**) $\phi^{(6)}$ (*line*), $\phi^{(7)}$ (*dot-dash*) and $\phi^{(8)}$ with $\alpha = 1$ (*dash*)

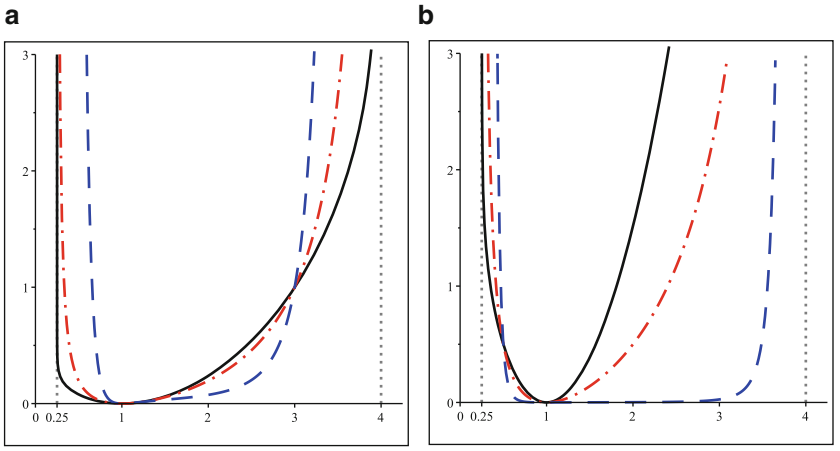


Fig. 3 Function $\phi^{(8)}(g;l,u,\alpha)$ for various values of α with $l = 1/4$ and $u = 4$. (**a**) $\phi^{(8)}$ scaled so that $c_{8,\alpha}\phi^{(8)}(3;l,u,\alpha) = 1$: $\alpha = 0.2$ (*line*), $\alpha = 1$ (*dot-dash*) and $\alpha = 5$ (*dash*). (**b**) $\phi^{(8)}$ scaled so that $c_{8,\alpha}\phi^{(8)}(\frac{1}{2};l,u,\alpha) = \frac{1}{2}$: $\alpha = 0.2$ (*line*), $\alpha = 1$ (*dot-dash*) and $\alpha = 5$ (*dash*)

we thus plot scaled versions of these functions using appropriate multiples (so that different functions become visually comparable).

We distinguish the following two types of functions ϕ:

Type I $\phi(g)$ is defined for all g either in $\mathbb{R}$ or $\mathbb{R}_+ = (0,\infty)$ and does not depend on l and u.

Type II $\phi(g)$ is defined for $g \in (l,u)$ but not outside the interval $[l,u]$. The functional form of g depends on l and u and hence we will use the notation $\phi(g;l,u)$ for the functions ϕ of this type.

The authors of the classical papers [4] and [5] suggest six choices for the function ϕ. Five of these are Type I and are $\phi^{(1)}(g) = (g-1)^2$, $\phi^{(2)}(g) = g\ln g - g + 1$, $\phi^{(3)}(g) = (\sqrt{g}-1)^2$, $\phi^{(4)}(g) = -\ln g + g - 1$ and $\phi^{(5)}(g) = (g-1)^2/g$. Figure 1 shows the shapes of these five functions.

The function $\phi^{(1)}$ is simply quadratic; in the literature on calibration it is usually referred to as the "chi-square" function (see for example [14, equation (2.10)]). It is by far the most popular in practice. The function $\phi^{(2)}$ is often referred to as the multiplicative or raking function in literature, (see for example [1]).

Many authors consider solving optimization problem (3) without the constraint $L \le G \le U$. However, in this case using the function $\phi^{(1)}$ in the optimization may lead to extreme and negative weights. Whilst the function $\phi^{(2)}$, by the nature of its domain, only permits non-negative values for the optimized weights, the weights may still take very large values. This also applies to functions $\phi^{(3)}$, $\phi^{(4)}$ and $\phi^{(5)}$. The functions $\phi^{(3)}$, $\phi^{(4)}$ and $\phi^{(5)}$ have received much less attention in the literature on calibration.

The above criticism of the functions $\phi^{(1)}$–$\phi^{(5)}$ can be extended to all functions of Type I. Note that if we use the functions ϕ of Type I then optimization problem (3)

is an optimization problem with many variables and many constraints (recall that n is typically very large).

Let us consider three functions ϕ of Type II:

$$\phi^{(6)}(g;l,u) = (g-l)\ln\left(\frac{g-l}{1-l}\right)+(u-g)\ln\left(\frac{u-g}{u-1}\right),$$

$$\phi^{(7)}(g;l,u) = (1-l)\ln\left(\frac{1-l}{g-l}\right)+(u-1)\ln\left(\frac{u-1}{u-g}\right), \tag{6}$$

$$\phi^{(8)}(g;l,u,\alpha) = \frac{(g-1)^2}{[(u-g)(g-l)]^{\alpha}}, \quad \alpha>0. \tag{7}$$

In Fig. 2, we plot the functions $c_1\phi^{(1)}(g)$, $c_6\phi^{(6)}(g;\frac{1}{4},4)$, $c_7\phi^{(7)}(g;\frac{1}{4},4)$ and $c_{8,1}\phi^{(8)}(g;\frac{1}{4},4,1)$ with the constants c_1, c_6, c_7 and $c_{8,1}$ chosen so that $c_1\phi^{(1)}(3)=1$, $c_k\phi^{(k)}(3;\frac{1}{4},4)=1$ for $k=6,7$ and $c_{8,1}\phi^{(8)}(3;\frac{1}{4},4,1)=1$.

In Fig. 3, we plot function $\phi^{(8)}$ for various values of the parameter α. In Fig. 3a, we choose the constants $c_{8,\alpha}$ so that $c_{8,\alpha}\phi^{(8)}(3;\frac{1}{4},4,\alpha)=1$. In Fig. 3b, we choose the constants $c_{8,\alpha}$ so that $c_{8,\alpha}\phi^{(8)}(\frac{1}{2};\frac{1}{4},4,\alpha)=\frac{1}{2}$.

The function $\phi^{(6)}$ is defined on the closed interval $g\in[l,u]$ so that by continuity we have $\phi^{(6)}(l;l,u)=(u-l)\ln\frac{u-l}{u-1}$ and $\phi^{(6)}(u;l,u)=(u-l)\ln\frac{u-l}{1-l}$. The function $\phi^{(6)}(g;l,u)$ is not defined outside the interval $[l,u]$. Using this function in (1) creates difficulties for the algorithms that optimize the function (1) because of the discontinuity (and the loss of convexity) of $\phi^{(6)}(g;l,u)$ at $g=l$ and $g=u$. A way around this is the use of constrained optimization algorithms but then the criticism above directed to the functions of Type I can be extended to the function $\phi^{(6)}$.

The functions $\phi^{(7)}(g;l,u)$ and $\phi^{(8)}(g;l,u,\alpha)$ are derived by us. These two functions are defined only in the open interval $g\in(l,u)$ and tend to infinity as g tends to either l or u so that they can be classified as interior penalty functions. We have derived the expression for the function $\phi^{(7)}$ by applying a suitable transformation (including taking a logarithm) to the density of the Beta-distribution on $[0,1]$. The convexity of the function $\phi^{(7)}$ follows from the expression for its second derivative:

$$\frac{\partial^2\phi^{(7)}(g;l,u)}{\partial g^2} = \frac{(u-l)\left(g^2-lu-2g+l+u\right)}{(g-l)^2(u-g)^2}$$
$$= \frac{(u-l)\left[(g-1)^2+(u-1)(1-l)\right]}{(g-l)^2(u-g)^2}.$$

Since $0 < l < 1 < u < \infty$, this second derivative is positive for all $g \in (l,u)$ so that the function $\phi^{(7)}(g;l,u)$ is convex. The analytic forms of the functions $\phi^{(6)}$ and $\phi^{(7)}$ are very similar but we believe the properties of the function $\phi^{(7)}$ are much more attractive for the problem at hand than the properties of the function $\phi^{(6)}$.

For any $\alpha > 0$, the function $\phi^{(8)}$ has properties similar to the function $\phi^{(7)}$: it is defined in the open interval $g \in (l,u)$, it is convex in this interval and it tends to infinity as $g \to l$ or $g \to u$. The function $\phi^{(8)}$ depends on an extra shape parameter α, see Fig. 3, so that the penalty for g deviating from 1 can be adjusted by the user.

A very important special case of the function $\phi^{(8)}$ occurs when $\alpha = 1$:

$$\phi^{(8)}(g;l,u,1) = \frac{(g-1)^2}{(u-g)(g-l)}. \tag{8}$$

The most attractive property of the function $\phi^{(8)}$ is its invariance with respect to the change $g \leftrightarrow 1/g$ in the case $l = 1/u$ (which is a very common case in practice). Recall that $g = w/d$ is the ratio of the calibrated weight w to the initial weight d and therefore the multiplicative scale for measuring deviations of g from 1 is the most appropriate. This means that it is very natural to penalize g as much as $1/g$ for deviating from 1. Assuming $\alpha = 1$ and $l = 1/u$ we have

$$\phi(g;u) = \phi^{(8)}(g;1/u,u,1) = \frac{(g-1)^2}{(u-g)(g-1/u)}.$$

For this function, we have $\phi(g;u) = \phi(1/g;u)$ so that this function possesses the additional property of equally penalizing g and $1/g$.

4 Hard Calibration

In Sect. 2, we introduced the calibration problem with both hard and soft constraints. In this section we consider optimization problem (3), namely calibration with hard constraints. We shall refer to this class of calibration problems as hard calibration. For several examples, we shall compare the calibrated weights obtained using each of the functions considered in Sect. 3.

We solve optimization problem (3) using the "solnp" function within R's Rsolnp package (see [6]). Using this software, we directly solve optimization problem (3) using the Augmented Lagrange Multiplier (ALM) method (see [9] for more details) for any choice of Type I or Type II function. For a comprehensive optimization software guide, see [12].

We consider two approaches to the hard calibration problem. The first of these is the classical approach considered in [4]. For this approach, the constraint $L \leq G \leq U$ is not included within the optimization. This means negative and extreme weights are in the domain of the feasible solution. This motivates the second approach that

considers the optimization problem (3) including the constraint $L \leq G \leq U$. The classical approach can be considered a particular case of the second approach, with L and U chosen to be vectors whose entries are $l = -\infty$ and $u = \infty$, respectively.

We remark that there are software packages that solve the calibration problem using an iterative Newton method as detailed in [5]. Examples of these include the "calib" function within R's sampling package (see [19]), the G-CALIB-S module within SPSS (see [20]) and the package CALMAR in SAS (see [4]). These packages allow the user to solve the hard calibration problem using the classical approach (no constraint $L \leq G \leq U$) for the functions $\phi^{(1)}$ and $\phi^{(2)}$. The packages also allow the user to solve the hard calibration problem including the constraint $L \leq G \leq U$ for functions $\phi^{(1)}$ and $\phi^{(6)}$ (see [5] for more details).

Many statistical offices throughout Europe use these packages to perform calibration. When comparing the weights obtained using direct optimization with the weights given by these packages, the answers in our examples were the same to within computer error (despite the running time was in some cases very different). Therefore, for the remainder of this chapter, we only solve optimization problem (3) using the ALM method.

To illustrate the case of negative and extreme weights, we consider the following example adapted from [8] using data from [3].

4.1 Example 1: A Classical Example

Throughout this example, we are working in units of thousands of people. Suppose we have a sample of $n = 12$ cities, sampled from 49 possible cities. We wish to weight our sample of cities appropriately to estimate the population total of the 49 cities.

For the 12 sampled cities, we know their size in 1920. Suppose we also know the population total of the 49 cities in 1920, namely $T = 5{,}054$. We begin with the vector $G = \mathbf{1}$ and take the initial weights $D = (49/12, 49/12, \ldots, 49/12)'$. These initial weights are derived using the classical Horvitz–Thompson estimator [7].

Recall from Sect. 2, that the hard calibration constraint can be written in the form $X'W = T$ or equivalently $A'G = T$, with $a_{ij} = d_i x_{ij}$. We only have one auxiliary variable in this example, thus X and A reduce to 12×1 vectors. Suppose we are given the sample values for the auxiliary variable in the 12×1 vector X, where $X = (93, 77, 61, 87, 116, 2, 30, 172, 36, 64, 66, 60)'$. Note that in this case $X'D = A'\mathbf{1} = 3528 \neq 5054$. Therefore, for the initial weights $G = \mathbf{1}$, the constraint $A'G = T$ is not satisfied. This motivates the need to calibrate.

Figure 4 shows the g-weights obtained when optimizing (3) for functions $\phi^{(1)}$, $\phi^{(2)}$ and $\phi^{(3)}$ using classical hard calibration (recall L and U are taken as vectors whose entries are $-\infty$ and ∞, respectively). We consider the case $q_i = d_i$ in (1). Figure 4a shows the calibrated weights when we do not impose the constraint $\mathbf{1}'G = 12$. The calibrated weights obtained when we impose the constraint $\mathbf{1}'G = 12$ are shown in Fig. 4b.

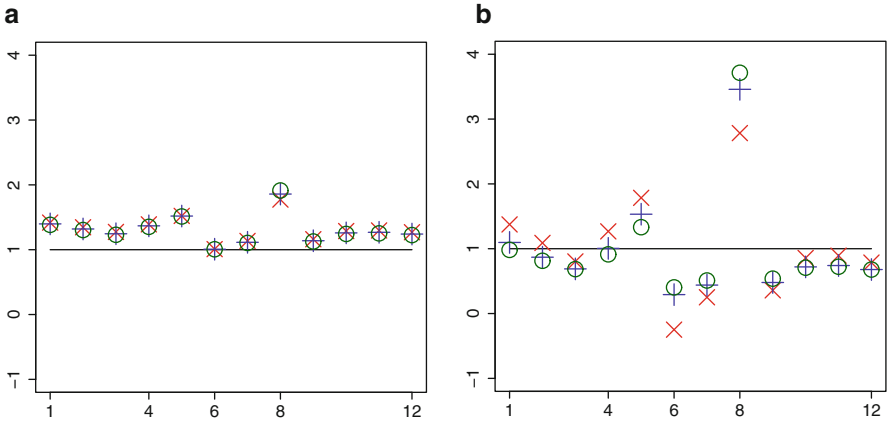


Fig. 4 Comparison of g-weights with $\mathbf{1}$ (*line*) for the functions $\phi^{(1)}$, $\phi^{(2)}$ and $\phi^{(3)}$. (**a**) g-weights for functions $\phi^{(1)}$ (*cross*), $\phi^{(2)}$ (*plus*) and $\phi^{(3)}$ (*circle*) with the sum of weights unconstrained. (**b**) g-weights for functions $\phi^{(1)}$ (*cross*), $\phi^{(2)}$ (*plus*) and $\phi^{(3)}$ (*circle*) with the sum of weights constrained

For these functions, observe that when we do not impose the constraint $\mathbf{1}'G = 12$, all the weights increase from, or remain at, their initial value of 1. It can be verified that the calibrated weights for each of these functions satisfy the constraint $A'G = T$. We remark that $\mathbf{1}'G = 15.883$ for the calibrated weights using function $\phi^{(1)}$, $\mathbf{1}'G = 15.738$ for the calibrated weights using function $\phi^{(2)}$ and $\mathbf{1}'G = 15.653$ for the calibrated weights using function $\phi^{(3)}$; in all cases $\mathbf{1}'G > 12$ due to the calibrated weights being larger than the initial weights of 1.

Imposing the extra constraint $\mathbf{1}'G = 12$ results in weights that are distributed both above and below the initial g-weights of $\mathbf{1}$. One of the g-weights for function $\phi^{(1)}$ (indexed 6 in Fig. 4b) is negative, whilst the weight indexed 8 has taken a large value in comparison to the other g-weights. For functions $\phi^{(2)}$ and $\phi^{(3)}$, we do not have a negative weight at index 6; however, the value of the weight at index 8 is still large in comparison with the other weights. Thus, whilst functions $\phi^{(2)}$ and $\phi^{(3)}$ prevent negative weights, they do not prevent large positive weights.

We remark that the behaviour of the weights for functions $\phi^{(4)}$ and $\phi^{(5)}$ is very similar to that for functions $\phi^{(2)}$ and $\phi^{(3)}$. Plots of the weights comparing functions $\phi^{(1)}$, $\phi^{(4)}$ and $\phi^{(5)}$ are very similar to the plots in Fig. 4a, b. Hence we do not plot the weights for functions $\phi^{(4)}$ and $\phi^{(5)}$ here.

To overcome the issue of negative and extreme weights, we include constraint $L \leq G \leq U$, where L and U have entries l and u, respectively, with $0 \leq l < 1 < u \leq \infty$. Any feasible solution to this problem is guaranteed to be within the bounds pre-specified by the user. However, recall from Sect. 2 that the feasible solution of this problem may be empty depending on the choice of L and U.

Returning to the example, suppose the calibrated weights G must satisfy the bounds $L \leq G \leq U$ where $L = (l, l, \ldots, l)'$ and $U = (u, u, \ldots, u)'$ are both 12×1

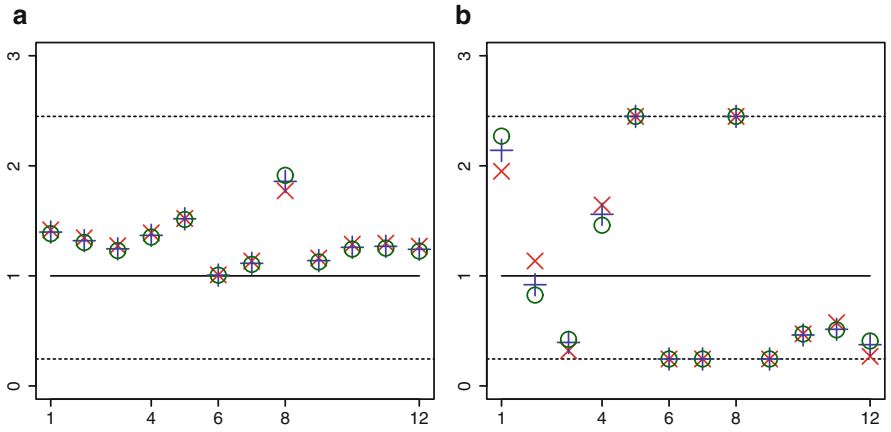


Fig. 5 Comparison of g-weights with $\mathbf{1}$ (*line*) for the functions $\phi^{(1)}$, $\phi^{(2)}$ and $\phi^{(3)}$, *dotted lines* indicate the upper and lower bounds. (**a**) g-weights for functions $\phi^{(1)}$ (*cross*), $\phi^{(2)}$ (*plus*) and $\phi^{(3)}$ (*circle*) with the sum of weights unconstrained. (**b**) g-weights for functions $\phi^{(1)}$ (*cross*), $\phi^{(2)}$ (*plus*) and $\phi^{(3)}$ (*circle*) with the sum of weights constrained

vectors. Consider the particular case of $l = \dfrac{12}{49}$ and $u = \dfrac{120}{49}$. This means the g-weights g_i will be bounded between the lower bound of $\dfrac{12}{49}$ and the upper bound of $\dfrac{120}{49}$, whilst the weights w_i will be bounded between the lower bound of $ld_i = 1$ and the upper bound of $ud_i = 10$ for all i.

Figure 5 shows the g-weights obtained by optimizing (3) for functions $\phi^{(1)}$, $\phi^{(2)}$ and $\phi^{(3)}$. Figure 5a shows the calibrated weights when we do not impose the constraint $\mathbf{1}'G = 12$. Figure 5b shows the calibrated weights when we include this constraint.

For the weights in Fig. 5a, we observe that imposing the constraint $\mathbf{1}'G = 12$ results in all the weights increasing from, or remaining at, their initial value of 1. The weights in Fig. 5a are identical to those in Fig. 4a.

However, in Fig. 5b, we see that imposing the extra constraint $\mathbf{1}'G = 12$ results in weights that are at, or very close to, the upper and lower bounds u and l, respectively. The weights in Fig. 5b are different from those in Fig. 4b.

In this case, the behaviour of the weights for functions $\phi^{(4)}$ and $\phi^{(5)}$ is very similar to that for functions $\phi^{(2)}$ and $\phi^{(3)}$, both with and without the constraint $\mathbf{1}'G = 12$ included in the optimization. Hence we do not plot the weights for these functions here.

Recall the relationship $w_i = d_i g_i$. Since the vector of initial weights D is given, and we have calculated the g-weights, we can compute the weights w_i. Computing the weights w_i for function $\phi^{(1)}$ from the corresponding g-weights in Fig. 5b gives the same weights as those derived in [8].

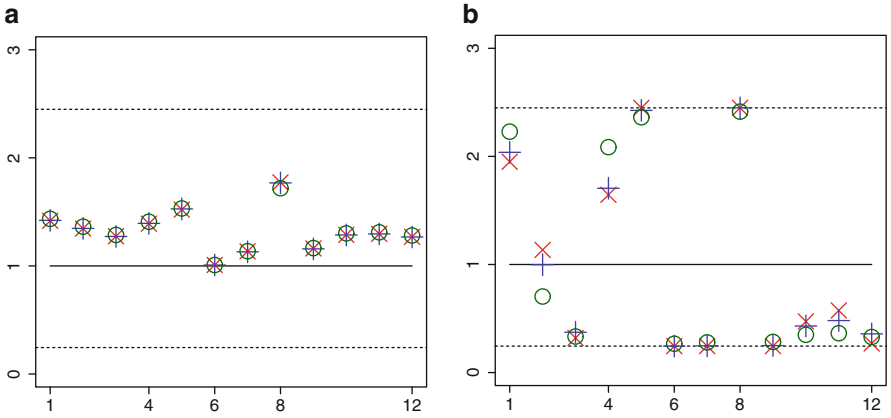


Fig. 6 Comparison of g-weights with $\mathbf{1}$ (*line*) for functions $\phi^{(1)}$, $\phi^{(6)}$ and $\phi^{(7)}$, *dotted lines* indicate the upper and lower bounds. (**a**) g-weights for functions $\phi^{(1)}$ (*cross*), $\phi^{(6)}$ (*plus*) and $\phi^{(7)}$ (*circle*) with the sum of weights unconstrained. (**b**) g-weights for functions $\phi^{(1)}$ (*cross*), $\phi^{(6)}$ (*plus*) and $\phi^{(7)}$ (*circle*) with the sum of weights constrained

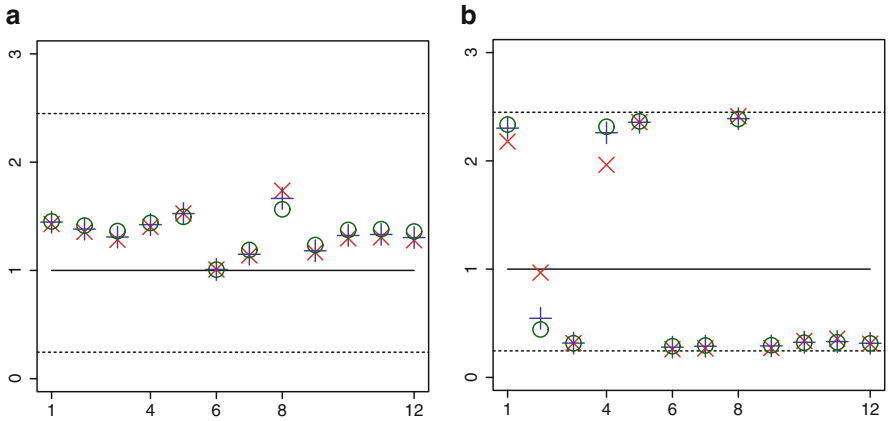


Fig. 7 Comparison of g-weights with $\mathbf{1}$ (*line*) for function $\phi^{(8)}$, *dotted lines* indicate the upper and lower bounds. (**a**) g-weights for function $\phi^{(8)}$ with $\alpha = 0.2$ (*cross*), $\alpha = 1$ (*plus*) and $\alpha = 5$ (*circle*) with the sum of weights unconstrained. (**b**) g-weights for function $\phi^{(8)}$ with $\alpha = 0.2$ (*cross*), $\alpha = 1$ (*plus*) and $\alpha = 5$ (*circle*) with the sum of weights constrained

Figure 6 shows the g-weights obtained by optimizing (3) for the functions $\phi^{(1)}$, $\phi^{(6)}$ and $\phi^{(7)}$. Figure 6a shows the calibrated weights when we do not impose the constraint $\mathbf{1}'G = 12$. Figure 6b shows the calibrated weights when the constraint is included within the optimization.

Figure 7 shows the g-weights obtained by optimizing (3) for function $\phi^{(8)}$ with α chosen to be 0.2, 1 and 5. Figure 7a shows the calibrated weights when we do not impose the constraint $\mathbf{1}'G = 12$. Figure 7b shows the calibrated weights when we include this constraint within the optimization.

Observe that when the constraint $\mathbf{1}'G = 12$ is not imposed, the weights all increase or remain at the initial values of 1. When the constraint is imposed, we see that the weights are distributed both above and below the initial values of 1, with several weights clustered at the bounds.

In summary, we have seen that not imposing the constraint $\mathbf{1}'G = 12$ results in calibrated weights exhibiting less variability than the calibrated weights obtained including the constraint. For this example, the calibrated weights all increased from the initial values of 1 but did not exhibit any extremal behaviour, lying well within the considered bounds. However, including the constraint $\mathbf{1}'G = 12$ gave calibrated weights that were more variable and likely to move towards the boundaries.

For the remaining examples in this chapter, we shall explore the effects the choice of L and U have on the calibrated weights G. In all the examples we will include the constraint $\mathbf{1}'G = n$, and take $q_i = d_i$ in (1).

4.2 Example 2

Suppose we are given the vector $X = (93, 77, 87, 116, 2, 30, 172, 36, 64, 60)'$ and the 10×1 vector of initial weights $D = (4, 4, \ldots, 4)'$. The parameter value $T = 3{,}900$ is assumed known. Recall that we impose the upper and lower bounds $U = (u, u, \ldots, u)'$ and $L = (l, l, \ldots, l)'$, where U and L are both 10×1 vectors whose entries are u and l, respectively. Consider the case $l = 1/u$. We wish to find the smallest value of u such that optimization problem (3) has a feasible solution. In this example, experimentation gave the smallest value of u as approximately 2.0.

In Fig. 8 we plot the calibrated weights when we take $l = 1/2$ and $u = 2$. In this case, solving optimization problem (3) for functions $\phi^{(1)}$, $\phi^{(6)}$ and $\phi^{(7)}$ gives the weights in Fig. 8a. Figure 8b shows the weights for function $\phi^{(8)}$ with $\alpha = 0.2$, $\alpha = 1$ and $\alpha = 5$.

For this example, a feasible solution to problem (3) exists for the (approximate) bounds $0 \leq l \leq 1/2$ and $u \geq 2$. Let us consider the effect of changing the values of l and u.

Figure 9 shows the calibrated weights when $l = 1/4$ and $u = 2$. In Fig. 9a we plot the weights for functions $\phi^{(1)}$, $\phi^{(6)}$ and $\phi^{(7)}$ whilst in Fig. 9b we plot the weights for function $\phi^{(8)}$ with $\alpha = 0.2$, $\alpha = 1$ and $\alpha = 5$. We see that reducing the lower bound results in less weights taking values at the lower bound. Generally, the calibrated weights for function $\phi^{(8)}$ appear to move towards the boundaries more than the weights obtained for functions $\phi^{(1)}$, $\phi^{(6)}$ and $\phi^{(7)}$.

We now consider the effect of increasing u. In Fig. 10, we keep $l = 1/4$ and consider the calibrated weights when $u = 4$. In Fig. 10a we plot the calibrated weights for the functions $\phi^{(1)}$, $\phi^{(6)}$ and $\phi^{(7)}$ whilst in Fig. 10b we plot the calibrated weights for function $\phi^{(8)}$ with $\alpha = 0.2$, $\alpha = 1$ and $\alpha = 5$. We see that increasing the upper bound has resulted in some of the weights increasing slightly in comparison to the weights in Fig. 9. However, there are no weights on the upper bound.

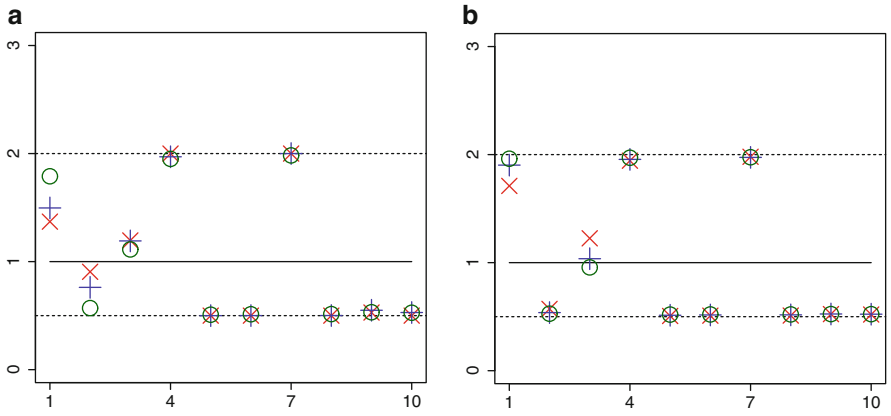


Fig. 8 Comparison of weights for functions $\phi^{(1)}$, $\phi^{(6)}$, $\phi^{(7)}$, and $\phi^{(8)}$ for various α with $l = 1/2$ and $u = 2$ (*dotted lines* indicate bounds). (**a**) Weights obtained for $\phi^{(1)}$ (*cross*), $\phi^{(6)}$ (*plus*) and $\phi^{(7)}$ (*circle*). (**b**) Weights obtained for $\phi^{(8)}$ with $\alpha = 0.2$ (*cross*), and $\alpha = 1$ (*plus*) and $\alpha = 5$ (*circle*)

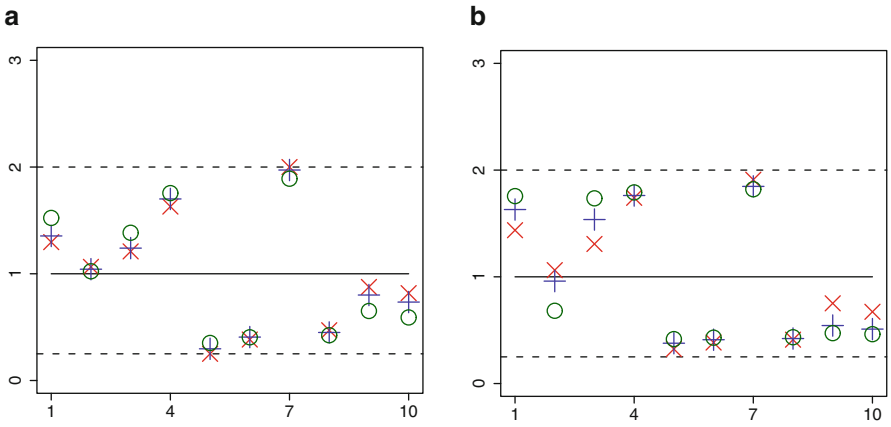


Fig. 9 Comparison of weights for functions $\phi^{(1)}$, $\phi^{(6)}$, $\phi^{(7)}$, and $\phi^{(8)}$ for various α with $l = 1/4$ and $u = 2$ (*dotted lines* indicate bounds). (**a**) Weights obtained for $\phi^{(1)}$ (*cross*), $\phi^{(6)}$ (*plus*) and $\phi^{(7)}$ (*circle*). (**b**) Weights obtained for $\phi^{(8)}$ with $\alpha = 0.2$ (*cross*), and $\alpha = 1$ (*plus*) and $\alpha = 5$ (*circle*)

To conclude, this example has shown that taking $l = 1/u$ and minimizing the value of u such that the calibration problem (3) has a feasible solution often results in many of the weights taking values at the boundaries. Increasing the value of u gives extra freedom to the weights and, as a result, there are typically less weights at the boundaries.

In the remaining two examples, we only consider the smallest value of u for which the optimization problem (3) has a feasible solution when $l = 1/u$. We further explore the phenomenon of weights clustering at the boundary and investigate whether different functions are more or less likely to give weights that approach the boundaries.

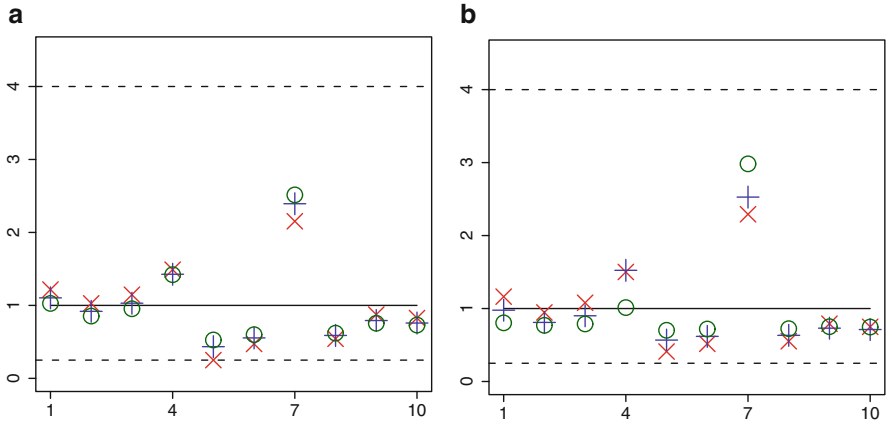


Fig. 10 Comparison of weights for functions $\phi^{(1)}$, $\phi^{(6)}$, $\phi^{(7)}$, and $\phi^{(8)}$ for various α with $l = 1/4$ and $u = 4$ (*dotted lines* indicate bounds). (**a**) Weights obtained for $\phi^{(1)}$ (*cross*), $\phi^{(6)}$ (*plus*) and $\phi^{(7)}$ (*circle*). (**b**) Weights obtained for $\phi^{(8)}$ with $\alpha = 0.2$ (*cross*), and $\alpha = 1$ (*plus*) and $\alpha = 5$ (*circle*)

4.3 Example 3

Suppose we are given the 100×1 vector of initial weights $D = (5, \ldots, 5)'$ and suppose that $T = 49{,}500$. The vector of auxiliary values X is formed by extending the auxiliary vector from Example 4.2. We form a 100×1 vector that has the values from the auxiliary vector in Example 4.2 as its first ten entries. The next ten entries are formed by taking the auxiliary vector from Example 4.2 and adding 2 to each value. In a similar way, we subtract 3 from each value to give the next ten values. In a similar way, we then repeat the vector, add 4 to all the entries, add 3 to all the entries, subtract 1, subtract 2, repeat the vector and finally add 4 to give the remaining 70 values.

We impose the upper and lower bounds $U = (u, u, \ldots, u)'$ and $L = (l, l, \ldots, l)'$, where L and U are both 100×1 vectors whose entries are u and $l = 1/u$, respectively. For this example, experimentation gives the smallest value of u as approximately $u = 2$ and so $l = 1/2$.

In Fig. 11, we compare the calibrated weights for functions $\phi^{(6)}$, $\phi^{(7)}$ and $\phi^{(8)}$ with those for function $\phi^{(1)}$. In Fig. 11a, we observe that most of the points in the scatterplot are on the diagonal. This indicates the similarity of the weights for functions $\phi^{(1)}$ and $\phi^{(6)}$. However, in Fig. 11b, we observe that there are fewer weights on the diagonal. This indicates that, for function $\phi^{(7)}$, more of the weights approach the boundary. In Fig. 11c, we see this even more clearly with a distinct band of weights at the upper and lower bounds of 2 and $\frac{1}{2}$ for function $\phi^{(8)}$, compared with the weights for $\phi^{(1)}$ that are more evenly distributed between the upper and lower bounds.

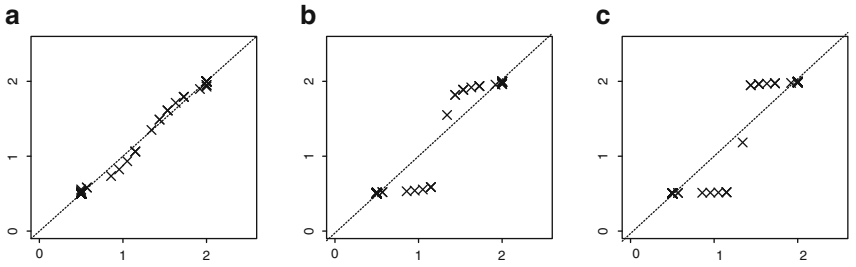


Fig. 11 Comparison of weights for function $\phi^{(1)}$ against functions $\phi^{(6)}$, $\phi^{(7)}$ and $\phi^{(8)}$, with $l = 1/2$ and $u = 2$. (**a**) Weights for function $\phi^{(1)}$ against $\phi^{(6)}$. (**b**) Weights for function $\phi^{(1)}$ against $\phi^{(7)}$. (**c**) Weights for function $\phi^{(1)}$ against $\phi^{(8)}$

For the next example, we keep the sample size at $n = 100$ and increase the number of auxiliary variables to $m = 3$.

4.4 Example 4

Suppose we are given a 100×1 vector of initial weights $D = (5, 5, \ldots, 5)'$, and let $T = (49500, 49540, 41000)'$. Suppose that the 100×3 matrix of auxiliary values X is defined as follows: for the first column of X we take the auxiliary vector from Example 3 in Sect. 4.3. For the second column of X, we form a 100×1 vector whose first ten values are formed by taking the auxiliary vector in Example 4.2 and subtracting 1. The next ten entries are formed by adding one to each of the values of the auxiliary vector in Example 4.2. In a similar way, we subtract 2 from each value to give the next ten values, then repeat the vector, add 5 to all the entries, repeat the vector twice, subtract 1, add 1 and finally add 3 to give the remaining 70 values. For the third column, we take 100 values generated at random from a Normal distribution with mean 80 and standard deviation 48 (these are similar to the mean and standard deviations for the other columns).

We impose the upper and lower bounds $U = (u, u, \ldots, u)'$ and $L = (l, l, \ldots, l)'$, where L and U are both 100×1 vectors whose entries are u and $l = 1/u$ respectively. For this example, experimentation gives the smallest value of u as approximately $u = 2$, and so $l = 1/2$.

In Fig. 12, we compare the calibrated weights using function $\phi^{(1)}$ with the calibrated weights for functions $\phi^{(6)}$, $\phi^{(7)}$ and $\phi^{(8)}$ ($\alpha = 1$).

Figure 12 has many similarities with Fig. 11 in Example 4.3. We observe that the weights for functions $\phi^{(1)}$ and $\phi^{(6)}$ are very similar. However, the calibrated weights for functions $\phi^{(7)}$ and $\phi^{(8)}$ show clear differences to the calibrated weights for function $\phi^{(1)}$. Again, we observe the distinct band of weights at the upper and

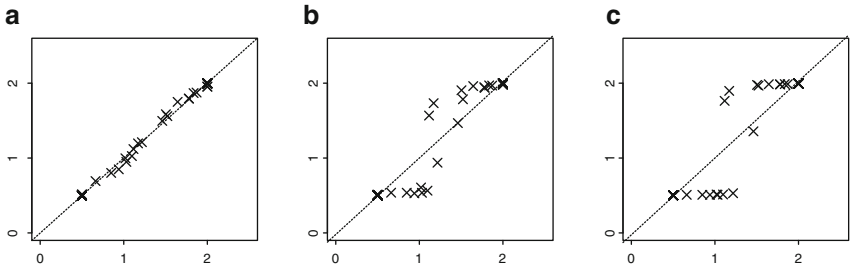


Fig. 12 Comparison of weights for the function $\phi^{(1)}$ against $\phi^{(6)}$, $\phi^{(7)}$ and $\phi^{(8)}$ ($\alpha = 1$). (**a**) Weights for function $\phi^{(1)}$ against $\phi^{(6)}$. (**b**) Weights for function $\phi^{(1)}$ against $\phi^{(7)}$. (**c**) Weights for function $\phi^{(1)}$ against $\phi^{(8)}$ with $\alpha = 1$

Table 1 CPU times for various functions ϕ in solving optimization problem (3) in Example 4

Function	CPU time (s)
$\phi^{(1)}$	0.609
$\phi^{(6)}$	0.734
$\phi^{(7)}$	0.544
$\phi^{(8)}$ ($\alpha = 0.2$)	0.569
$\phi^{(8)}$ ($\alpha = 1$)	0.559

lower bounds of 2 and $\frac{1}{2}$ for functions $\phi^{(7)}$ and $\phi^{(8)}$, compared with the weights for functions $\phi^{(1)}$ and $\phi^{(6)}$ that are more evenly distributed between the upper and lower bounds.

We now compare the CPU times taken to obtain the weights in Fig. 12. These CPU times were computed on a computer with an Intel(R) Core(TM) i7-4500U CPU Processor with 8 GB of RAM. The CPU times are given in Table 1. We observe that the CPU times for functions $\phi^{(7)}$ and $\phi^{(8)}$ ($\alpha = 0.2$) are less than those for the classical functions $\phi^{(1)}$ and $\phi^{(6)}$. CPU time is related to the complexity of the optimization problem, see [13] on a comprehensive discussion of how to measure numerical complexity of an optimization problem.

In these examples, we have seen that problem (3) does not necessarily have a feasible solution for all choices of the vectors L and U. We address this issue in the next section by introducing soft calibration.

5 Soft Calibration

In this section, we consider optimization problem (5). Recall that this requires a choice of the functions Φ and Ψ. In this section, we choose Φ to be of the form (1) with ϕ taken to be $\phi^{(1)}$ and consider the penalty function Ψ of the form (2). We do not consider other choices of Φ or Ψ in this section.

Rewriting problem (5) with our choice of Φ and Ψ gives the following optimization problem:

$$\sum_{i=1}^{n} q_i(g_i - 1)^2 + \beta(A'G - T)'C(A'G - T) \to \min_{G \in \mathbb{G}}, \tag{9}$$

where $\mathbb{G} = \{G : L \leq G \leq U\}$, $q_1, \ldots, q_n$ are given non-negative numbers, C is a user-specified $m \times m$ positive definite (usually diagonal) matrix and $\beta > 0$ is some constant.

In Sect. 4, we considered two approaches to solving the hard calibration problem (3). We now consider two similar approaches for solving problem (9). The first approach is the classical soft calibration approach (see for example [2]). In this approach, the constraint $L \leq G \leq U$ is not included within the optimization. Practitioners vary the value of the parameter β so that the weights are within some pre-specified bounds. The second approach is to include the constraint $L \leq G \leq U$ within the optimization algorithm, i.e. to solve optimization problem (5). We remark that classical soft calibration is a special case of the second approach where L and U are vectors whose entries are $-\infty$ and ∞, respectively.

For the example in Sect. 4.1, we considered the calibrated weights obtained when solving optimization problem (3) without imposing the constraint $L \leq G \leq U$. In this case, we saw that it is possible to obtain negative and extreme weights.

The classical soft calibration problem was proposed as a way to deal with these negative and extreme weights. Classical soft calibration allows an analytic solution to be found to optimization problem (9). Let $\mathbb{D}$ be an $n \times n$ diagonal matrix, whose entries are the weights $d_1, d_2, \ldots d_n$. Furthermore, take $q_i = d_i$ and let $\gamma = \dfrac{1}{\beta}$. Then, for the classical soft calibration approach, the analytic form of the weights that satisfy optimization problem (9) is given by

$$G = \mathbf{1} + A\left(A'\mathbb{D}^{-1}A + \gamma C^{-1}\right)^{-1}\left(T - A'\mathbf{1}\right). \tag{10}$$

This is an equivalent formulation of equation (2.4) from [2], expressed in terms of g-weights. The term $\left(A'\mathbb{D}^{-1}A + \gamma C^{-1}\right)^{-1}$ is similar to the inverse matrix term in ridge regression (see for example [15]).

Let us consider the effect of changing the parameter β in (9). Recall that $\gamma = 1/\beta$ or equivalently $\beta = \gamma^{-1}$. We consider the effect of changing the parameter γ. As γ tends to zero, γ^{-1} tends to infinity and so optimization problem (9) reduces to minimising $(A'G - T)'C(A'G - T)$ for $G \in \mathbb{G}$. As this term is quadratic, the minimum occurs when $A'G - T = 0$ or equivalently $A'G = T$. This is the hard calibration constraint. Therefore, the case $\gamma \to 0$ corresponds to solving the hard calibration problem (3). We remark that this is consistent with (10), since taking $\gamma = 0$ in this formula gives the expression for the g-weights in classical hard calibration.

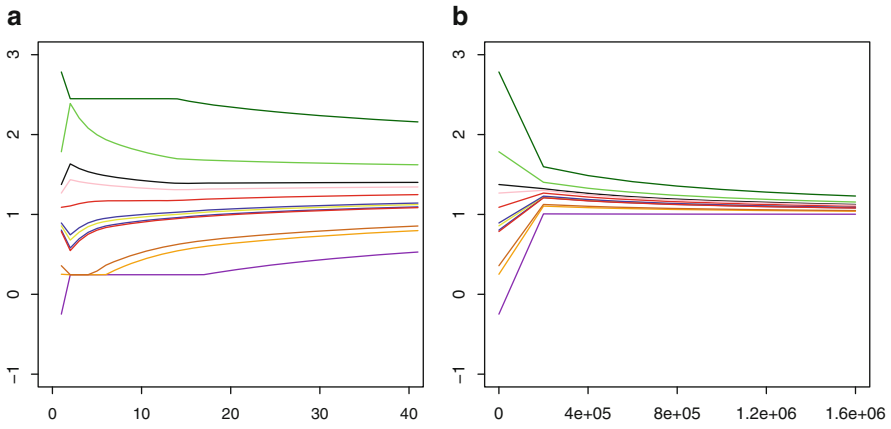


Fig. 13 Plots of classical soft calibration weights (10) as a function of γ. (**a**) Soft weights for γ between 0 and 40. (**b**) Soft weights for γ from 0 to 1.6×10^6

As γ tends to infinity, γ^{-1} tends to zero and so the term $(A'G-T)'C(A'G-T)$ becomes negligible. This results in optimization (9) reducing to the problem of minimizing $\Phi(G) = \sum_{i=1}^{n} q_i \phi^{(1)}(g_i)$ for $G \in \mathbb{G}$, which is minimized at $G = \mathbf{1}$ (by definition of the function Φ). Again, this is consistent with (10), since when $\gamma \to \infty$ the term $A\left(A'\mathbb{D}^{-1}A + \gamma C^{-1}\right)^{-1}\left(T - A'\mathbf{1}\right)$ tends to zero giving $G = \mathbf{1}$.

To illustrate this, let us revisit the example of Sect. 4.1. Recall that $T = 5{,}054$, $D = (49/12, 49/12, \ldots, 49/12)'$ and $X = (93, 77, 61, 87, 116, 2, 30, 172, 36, 64, 66, 60)'$. In Fig. 13, we plot the weights given by (10) as the value of γ varies. We take $C = I_m$, where I_m denotes the $m \times m$ identity matrix. Figure 13a plots the weights for values of γ from 0 to 40. This plot confirms our earlier assertions that as $\gamma \to 0$, G tends to the classical hard calibration weights. Figure 13b plots the weights for values of γ between 0 and 1.6×10^6. This plot confirms that as $\gamma \to \infty$, the g-weights tend to their initial values of 1.

When obtaining the explicit solution, (10), to the classical soft calibration problem, we did not specify any constraints on the weights G. Suppose that we wish to impose the constraint $L \leq G \leq U$. Observe from Fig. 13a that as the value of γ increases, the range of the weights decreases. In classical soft calibration, having obtained the analytic solution (10) for the calibrated weights, the approach to satisfying the constraint $L \leq G \leq U$ is to choose the smallest value of γ for which the weights in (10) are within the specified bounds. Clearly, the value of γ that satisfies the constraints $L \leq G \leq U$ is sample dependent.

Consider again Example 1 from Sect. 4.1. We previously saw that, in the case of classical hard calibration, we obtain negative and extreme weights for this sample. Suppose we wish to impose the lower and upper bounds of $l = 12/49$ and $u = 120/49$. We saw that we were able to satisfy these bounds by solving problem (3). In order to satisfy these bounds for classical soft calibration, experimentation gives

the smallest value of γ as approximately $\gamma = 9.0$ in order to find a solution that lies between these bounds. This is a relatively large value of γ.

Note that in this case we have $\mathbf{1}'G = 13.527 \neq 12$ and $A'G = 5053.899 \neq 5054$, therefore our constraints $\mathbf{1}'G = 12$ and $A'G = T$ are no longer satisfied. Having relaxed these constraints in the soft calibration penalty (2), the larger the value of γ, the smaller the value of β and the less importance we assign to the penalty (2) in (9). This allows greater variation between $A'G$ and T and between $\mathbf{1}'G$ and 12. However, for large values of γ there is less variation in the weights. In contrast, for small values of γ, the penalty (2) is given more importance allowing less variation between $A'G$ and T and between $\mathbf{1}'G$ and 12. However, in this case there will be greater variability in the weights.

We illustrate this in Figs. 14 and 15. To produce these figures, we took 10,000 simple random samples of size 12 from the data in [3]. Figure 14 shows the distribution of weights and values of $A'G$ when we take $\gamma = 0.1$. Figure 15 shows the distribution of weights and values of $A'G$ when we take $\gamma = 9$, as required for this example to ensure the weights are between L and U. We observe that although $\gamma = 9$ gave g-weights satisfying the bounds $L \leq G \leq U$ for one sample, this value of γ does not guarantee that the g-weights will satisfy these bounds for every sample.

Let us now consider the second approach of directly optimizing (5). As stated in Sect. 2, optimization problem (5) has a solution for any value of $\beta > 0$. Therefore, given any L and U, we can find a solution to optimization problem (5) independent of the choice of β. That is what makes this approach different to classical soft calibration.

Let us return again to Example 1 from Sect. 4.1. Consider the problem (5) with $L \leq G \leq U$ where $L = (l, \ldots, l)'$ and $U = (u, \ldots, u)'$ are 12×1 vectors with entries

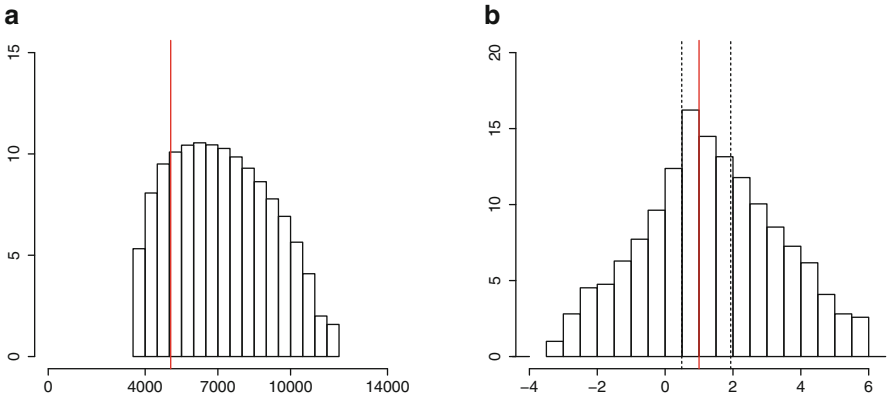


Fig. 14 Plots of $A'G$ and g-weights that satisfy optimization problem (10) for $\gamma = 0.1$. (**a**) $A'G$ for 10,000 random samples of size 12, *vertical line* at 5,054 ($A'G = 5{,}054$ is hard constraint). (**b**) g-weights for 10,000 random samples of size 12, *vertical line* at 1 (initial weights), *dashed lines* indicate bounds

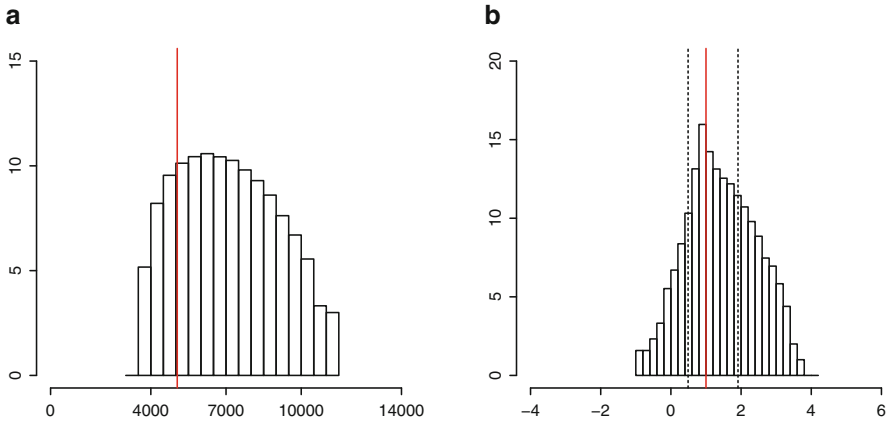


Fig. 15 Plots of $A'G$ and g-weights that satisfy the optimization problem (10) for $\gamma = 0.1$. (**a**) $X'W$ for 10,000 random samples of size 12, vertical line at 5,054 ($X'W = 5{,}054$ is hard constraint). (**b**) g-weights for 10,000 random samples of size 12, *vertical line* at 1 (initial weights), *dashed lines* indicate bounds

$l = \dfrac{12}{49}$ and $u = \dfrac{120}{49}$, respectively. We know that small values of γ give a solution that is close to the hard calibration solution. Taking $\gamma = 0.01$, we obtain soft calibration weights that are very similar to those derived for hard calibration in Sect. 4.1. Therefore, in this instance, solving problem (5) has little advantage over solving the corresponding hard calibration problem (3).

However, suppose we want to impose the bounds $l = 24/49$ and $u = 96/49$, corresponding to bounding the weights w_i between the lower and upper bounds of 2 and 8, respectively. In this case, there is no feasible solution to the hard calibration problem (3). Solving this problem using classical soft calibration requires a value of $\gamma = 16$ to ensure that the weights are between these bounds.

We now consider the direct optimization approach. Recall that for small values of γ, the solution to problem (5) is approximately equal to the solution to problem (3). Assuming we have the lower bounds $l = 24/49$ and $u = 96/49$, taking $\gamma = 10^{-9}$ we obtain weights G such that $A'G = 5{,}053.910$ and $\mathbf{1}'G = 13.435$. Under hard calibration, we would require $A'G = 5{,}054$ and $\mathbf{1}'G = 12$. We have almost satisfied the constraint $A'G = 5{,}054$; however, we have not satisfied the constraint $\mathbf{1}'G = 12$. This suggests that the condition $\mathbf{1}'G = 12$ was too restrictive.

6 Conclusions

The problem of calibrating weights in surveys is a very important practical problem. In the literature on calibration, there are many recipes but no clear understanding of what calibration is. In this chapter, we have formally formulated the calibration

problem as an optimization problem and defined the desired conditions for the components of the objective function and feasible region. We have demonstrated that the commonly used calibration criteria do not fully satisfy the desired criteria. The corresponding optimization problems are not flexible enough, harder than they have to be, or have some common recipes leading to wrong and contradictory recommendations. An example of the latter is the use of ridge estimators for trying to achieve positivity of the calibrated weights, see Sect. 5.

We have studied the influence of the function ϕ, the main component of objective function, on the complexity of the optimization problem and the final solution. We claim that the new functions $\phi^{(7)}$ and $\phi^{(8)}$ suggested in this chapter are much more transparent and more flexible than the functions adopted in the standard calibration literature and classical calibration software packages. The functions suggested by us lead to easier optimization problems as they automatically take into account the constraint $L \leq G \leq U$. This could be of high importance in practice as the dimension of the problem (which is the size of the sample) may be very large.

In the case of large samples, one of our recommendations is to replace the hard calibration problem defined by (1) and (3) with a soft calibration problem defined by (1), (2) and (5), where β in (2) is large and the functions ϕ_i in (1) are either $\phi_i^{(7)}$ or $\phi_i^{(8)}$, see (6) and (7), respectively. In doing so we replace a potentially difficult constrained optimization problem (3) with a much simpler problem (5), which is an unconstrained convex optimization problem (recall that all constraints in (5) are taken into account due to a clever choice of the functions ϕ_i). If β is large then the solution of this problem is guaranteed to be very close to the solution of the original problem (3).

References

1. Bankier, M., Houle, A.M., Luc, M.: Calibration estimation in the 1991 and 1996 Canadian censuses. In: Proceedings of the Survey Research Methods Section, pp. 66–75 (1997)
2. Bocci, J., Beaumont, C.: Another look at ridge calibration. Metron **66**(1), 5–20 (2008)
3. Cochran, W.G.: Sampling Techniques. Wiley, New York (1977)
4. Deville, J.C., Särndal, C.E.: Calibration estimators in survey sampling. J. Am. Stat. Assoc. **87**(418), 376–382 (1992)
5. Deville, J.C., Särndal, C.E., Sautory, O.: Generalized raking procedures in survey sampling. J. Am. Stat. Assoc. **88**(423), 1013–1020 (1993)
6. Ghalanos, A., Theussl, S.: Rsolnp: general non-linear optimization. R package version (2010)
7. Horvitz, D.G., Thompson, D.J.: A generalization of sampling without replacement from a finite universe. J. Am. Stat. Assoc. **47**(260), 663–685 (1952)
8. Huang, E., Fuller, W.: Nonnegative regression estimation for sample survey data. In: Proceedings of the Social Statistics Section, vol. 21, pp. 300–305. American Statistical Association, Alexandria (1978)
9. Ito, K., Kunisch, K.: Augmented lagrangian methods for nonsmooth, convex optimization in Hilbert spaces. Nonlinear Anal. Theory Methods Appl. **41**(5), 591–616 (2000)
10. Kott, P.S.: Using calibration weighting to adjust for nonresponse and coverage errors. Surv. Methodol. **32**(2), 133–142 (2006)

11. Lundström, S., Särndal, C.E.: Calibration as a standard method for treatment of nonresponse. J. Off. Stat. **15**, 305–327 (1999)
12. More, J.J., Wright, S.J., Pardalos, P.M.: Optimization Software Guide, vol. 14. Society for Industrial and Applied Mathematics, Philadelphia (1993)
13. Pardalos, P.M.: Complexity in Numerical Optimization. World Scientific, Singapore (1993)
14. Rao, J., Singh, A.: A ridge shrinkage method for range restricted weight calibration in survey sampling. In: Proceedings of the Section on Survey Research Methods, pp. 57–65 (1997)
15. Ryan, T.P.: Modern Regression Methods. Wiley, New York (2008)
16. Särndal, C.: The calibration approach in survey theory and practice. Surv. Methodol. **33**(2), 99–119 (2007)
17. Singh, S., Arnab, R.: A bridge between the GREG and the linear regression estimators. In: Joint Statistical Meeting, ASA Section on Survey Research Methods, Seattle, pp. 3689–3693 (2006)
18. Théberge, A.: Calibration and restricted weights. Surv. Methodol. **26**(1), 99–108 (2000)
19. Tillé, Y., Matei, A.: Rsolnp: General non-linear optimization. R package version (2013)
20. Vanderhoeft, C.: Generalised calibration at statistics Belgium: SPSS Module G-CALIB-S and current practices. Inst. National de Statistique (2001)

On the Sensitivity of Least Squares Data Fitting by Nonnegative Second Divided Differences

Ioannis C. Demetriou

Abstract Let measurements of a real function of one variable be given. If the function is convex but convexity has been lost due to errors of measurement, then we make the least sum of squares change to the data so that the second divided differences of the smoothed values are nonnegative. The underlying calculation is a quadratic programming algorithm and the piecewise linear interpolant to the solution components is a convex curve. Problems of this structure arise in various contexts in research and applications in science, engineering and social sciences. The sensitivity of the solution is investigated when the data are slightly altered. The sensitivity question arises in the utilization of the method. First some theory is presented and then an illustrative example shows the effect of specific as well as random changes of the data to the solution. As an application to real data, an experiment on the sensitivity of the convex estimate to the Gini coefficient in the USA for the time period 1947–1996 is presented. The measurements of the Gini coefficient are considered uncertain, with a uniform probability distribution over a certain interval. Some consequences of this uncertainty are investigated with the aid of a simulation technique.

Keywords Least squares • Data fitting • Quadratic programming • Piecewise linear interpolation • Sensitivity analysis

1 Introduction

The purpose of this chapter is to investigate the sensitivity of the least squares convex fit to discrete data with respect to changes in the data, where convexity enters in terms of nonnegative second divided differences of the data. Problems of this structure arise in various contexts, for example in estimating certain supply, demand and production relations in economics, where increasing returns (convexity)

I.C. Demetriou (✉)
Division of Mathematics and Informatics, Department of Economics, University of Athens, 1 Sofokleous and Aristidou Street, Athens 10559, Greece
e-mail: demetri@econ.uoa.gr

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_5

and diminishing returns (concavity) are assumed [14]. Other examples arise from estimating a utility function that is represented by a finite number of observations [16], from calculating the piecewise convex/concave approximation to discrete noisy data [6] and from determining the dopant profile in semiconductors [17], for instance.

We assume that the data come from an unknown underlying convex function $f(x)$, but convexity has been lost due to errors of measurement. The data are the coordinates $(x_i, \phi_i) \in \mathcal{R}^2$, for $i = 1, 2, \ldots, n$, where the abscissae satisfy the inequalities $x_1 < x_2 < \cdots < x_n$, ϕ_i is the measurement of $f(x)$ at x_i and, in view of our assumption, $\phi_i \approx f(x_i)$. We regard the measurements as components of a n-vector $\underline{\phi}$. Vectors will be considered column vectors unless the superscript "T" denotes transposition to a row vector.

Demetriou and Powell [5] studied the problem of calculating a n-vector $\underline{y}$ that minimizes the objective function

$$\Phi(\underline{y}) = \sum_{i=1}^{n} (\phi_i - y_i)^2 \tag{1}$$

subject to the convexity constraints

$$y[x_{i-1}, x_i, x_{i+1}] \geq 0, \; i = 2, 3, \ldots, n-1, \tag{2}$$

where

$$y[x_{i-1}, x_i, x_{i+1}] = \frac{y_{i-1}}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} + \frac{y_i}{(x_i - x_{i-1})(x_i - x_{i+1})} + \frac{y_{i+1}}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} \tag{3}$$

is the ith second divided difference on the components of $\underline{y}$ (see [2] for a definition). We call feasible any n-vector that satisfies the constraints (2). Since the constraints on $\underline{y}$ are linear, we introduce the scalar product notation

$$y[x_{i-1}, x_i, x_{i+1}] = \underline{a}_i^T \underline{y}, \; i = 2, 3, \ldots, n-1, \tag{4}$$

where $\underline{a}_i$, for $i = 2, 3, \ldots, n-1$ denote the constraint normals with respect to $\underline{y}$. By taking into account that each divided difference depends on only three adjacent components of $\underline{y}$, one can immediately see that the constraint normals are linearly independent vectors. In view of the strict convexity of (1) and the linearity and the consistency of constraints (2), the solution to this problem, say it is $\underline{y}^*$, is unique. Throughout the chapter we use occasionally the descriptive term *convex fit* for the solution.

The Karush–Kuhn–Tucker conditions (see, [8, p. 200]) provide necessary and sufficient conditions for optimality. They state that $\underline{y}^*$ is optimal if and only if

constraints (2) are satisfied and there exist nonnegative Lagrange multipliers $\{\lambda_i^* : i \in \mathcal{A}^*\}$ such that the first order conditions

$$\underline{y}^* - \underline{\phi} = \tfrac{1}{2} \sum_{i \in \mathcal{A}^*} \lambda_i^* \underline{a}_i \tag{5}$$

hold, where $\mathcal{A}^*$ is a subset of the constraint indices $\{2, 3, \ldots, n-1\}$ with the property

$$\underline{a}_i^T \underline{y}^* = 0, \quad i \in \mathcal{A}^*. \tag{6}$$

It is straightforward to calculate the solution of this optimization problem by standard strictly convex quadratic programming methods (see for example [8]), but a special version of the quadratic programming algorithm of Goldfarb and Idnani [10] has been developed by Demetriou and Powell [5] that is faster than general quadratic programming algorithms.

In Sect. 2 we give an outline of the algorithm for calculating $\underline{y}^*$ and we introduce notation, terminology and definitions that are needed for the presentation of this work. In Sect. 3 we give some results on sensitivity analysis by considering the sensitivity of the solution with respect to changes in the data. In Sect. 4 we give an example with as many as 12 measurements that make it easy to present and discuss the effect of specific as well as random changes to the solution. In Sect. 5 an experiment on the sensitivity of the convex estimate to the Gini coefficient measurements in the USA for the time period 1947–1996 is presented. The Gini coefficients are considered uncertain, with a uniform probability distribution over a certain interval. The consequences of this uncertainty are investigated with the aid of a simulation technique. In Sect. 6 we present some concluding remarks.

The numerical experiments were carried out by the Fortran software L2CXFT that has been written by Demetriou [3]. This package consists of about 1,600 lines including comments. The calculations were performed on a HP 8770w portable workstation with an Intel Core i7-3610QM, 2.3 GHz processor, which was used with the standard Fortran compiler of the Intel Visual Fortran Composer XE2013 in double precision arithmetic (first 15 decimal digits are significant) operating on Windows 7 with 64 bits word length.

2 An Outline of the Method of Calculation

In this section we outline the quadratic programming method of Demetriou and Powell [5] for calculating the solution of the problem of Sect. 1. We also introduce notation, terminology and definitions that are needed for presenting some results on sensitivity analysis. This method is by far faster than a general quadratic programming algorithm because it takes into account the structure of the constraints. A large part of its efficiency is due to a linear B-spline representation of the solution and the banded matrices that occur. For proofs, one may consult the above reference.

The method begins by calculating an initial approximation to the convex fit in only $O(n)$ computer operations, which is an advantage to the subsequent quadratic programming calculation because either it identifies set $\mathcal{A}^*$ or it comes quite close to it. The quadratic programming algorithm generates a finite sequence of subsets $\{\mathcal{A}^{(k)}:\ k=1,2,\ldots\}$ of the constraint indices $\{2,3,\ldots,n-1\}$ with the property

$$\underline{a}_i^T \underline{y} = 0,\ i \in \mathcal{A}^{(k)}. \tag{7}$$

For each k, we denote by $\underline{y}^{(k)}$ the vector that minimizes (1) subject to Eq. (7) and we call each constraint in (7) an active constraint. All the active constraints constitute the active set. Since the constraint normals are linearly independent, unique Lagrange multipliers $\{\lambda_i^{(k)}: i \in \mathcal{A}^{(k)}\}$ are defined by the first order optimality condition

$$\underline{y}^{(k)} - \underline{\phi} = \tfrac{1}{2} \sum_{i \in \mathcal{A}^{(k)}} \lambda_i^{(k)} \underline{a}_i. \tag{8}$$

Quadratic programming starts by deleting constraints if necessary from the active set derived by the mentioned $O(n)$ approximation until all the remaining active constraints have nonnegative Lagrange multipliers. This gives $\mathcal{A}^{(1)}$. If $\mathcal{A}^{(k)}$, for $k \geq 1$ is not set $\mathcal{A}^*$, then the quadratic programming algorithm adds to the active set the most violated constraint and deletes constraints with negative multipliers alternately, until the Karush–Kuhn–Tucker conditions are satisfied. Related to each $\mathcal{A}^{(k)}$, this process requires the calculation of $\underline{y}^{(k)}$ and $\underline{\lambda}^{(k)}$. Specifically, for each integer i in $\mathcal{A}^{(k)}$ we pick the ith row of (8) multiplied by $(x_{i-1} - x_{i+1})$, so the first or last row is never chosen, which gives a block tridiagonal positive definite system of equations. For example, if 2, 3 and 4, but not 5 are in $\mathcal{A}^{(k)}$, then the first of the blocks is

$$\begin{pmatrix} \dfrac{x_3 - x_1}{(x_2 - x_1)(x_3 - x_2)} & \dfrac{1}{x_3 - x_2} & 0 \\ \dfrac{1}{x_3 - x_2} & \dfrac{x_4 - x_2}{(x_3 - x_2)(x_4 - x_3)} & \dfrac{1}{x_4 - x_3} \\ 0 & \dfrac{1}{x_4 - x_3} & \dfrac{x_5 - x_3}{(x_4 - x_3)(x_5 - x_4)} \end{pmatrix}, \tag{9}$$

which is a positive definite matrix. It follows that $\underline{\lambda}^{(k)}$ can be derived efficiently and stably by a Cholesky factorization in only $O(|\mathcal{A}^{(k)}|)$ computer operations, where $|\mathcal{A}^{(k)}|$ is the number of elements of $\mathcal{A}^{(k)}$.

The equality constrained minimization problem of $\underline{y}^{(k)}$ forms an important part of the calculation, because it is solved very efficiently by a reduction to an equivalent unconstrained one with fewer variables due to a linear B-spline representation (for a definition see for example de Boor [2]). If $y(x)$, $x_1 \leq x \leq x_n$ is the piecewise linear interpolant to the points $\{(x_i, y_i^{(k)}):\ i = 1,2,\ldots,n\}$, then $y(x)$ has its knots on the set $\{x_i:\ i \in \{1,2,\ldots,n\} \setminus \mathcal{A}^{(k)}\}$ including x_1 and x_n. Indeed, the

equation $y^{(k)}[x_{i-1},x_i,x_{i+1}]=0$, when $i\in\mathcal{A}^{(k)}$ implies the collinearity of the points $(x_{i-1},y_{i-1}^{(k)})$, $(x_i,y_i^{(k)})$ and $(x_{i+1},y_{i+1}^{(k)})$, but if $y^{(k)}[x_{i-1},x_i,x_{i+1}]>0$, then i is the index of a knot of $y(x)$. Thus the knots of $y(x)$ are determined from the abscissae due to the active set. Let $j=n-1-|\mathcal{A}^{(k)}|$, let $\{\xi_p:\ p=1,\ldots,j-1\}$ be the interior knots of $y(x)$ in ascending order, let also $\xi_{-1}=\xi_0=x_1$ and $\xi_j=\xi_{j+1}=x_n$, and let $\{B_p:\ p=0,1,\ldots,j\}$ be a basis of normalized linear B-splines that are defined on $\{x_i:\ i=1,2,\ldots,n\}$ and satisfy the equations $B_p(\xi_p)=1$ and $B_p(\xi_q)=1$, $p\neq q$:

$$B_p(x)=\begin{cases}(x-\xi_{p-1})/(\xi_p-\xi_{p-1}), & \xi_{p-1}\leq x\leq\xi_p\\ (\xi_{p+1}-x)/(\xi_{p+1}-\xi_p), & \xi_p\leq x\leq\xi_{p+1}\\ 0, & \text{otherwise.}\end{cases}\tag{10}$$

Then $y(x)$ may be written uniquely in the form

$$y(x)=\sum_{p=0}^{j}\sigma_pB_p(x),\ x_1\leq x\leq x_n,\tag{11}$$

where the coefficients $\{\sigma_p:\ p=0,1,\ldots,j\}$ are the values of $y(x)$ at the knots and are calculated by solving the normal equations associated with the minimization of (1),

$$\sum_{p=0}^{j}[\sum_{i=1}^{n}B_k(x_i)B_p(x_i)]\sigma_p=\sum_{i=1}^{n}B_k(x_i)\phi_i,\ k=0,1,\ldots,j.\tag{12}$$

Since

$$\sum_{i=1}^{n}B_k(x_i)B_p(x_i)=0,\ \text{for}\ |k-p|>1,\tag{13}$$

system (12) simplifies to a positive definite tridiagonal system of equations, which can be solved for $\{\sigma_p:\ p=0,1,\ldots,j\}$ efficiently and stably by a Cholesky factorization in $O(j)$ computer operations. The intermediate components of $\underline{y}^{(k)}$ are found by linear interpolation to the spline coefficients due to (10) and (11). Further, the numerical results of [5] show that $|\mathcal{A}^{(k)}|$ in practice is usually large, so j is small, which is a major saving for the calculation.

3 Changes in the Solution Due to Changes in the Data

In this section we study the changes that occur in the solution of the minimization of function (1) subject to constraints (2) when the data are slightly altered. Stating formally, the *perturbed problem* requires to minimize the objective function

$$\tilde{\Phi}(\underline{y})=\underline{y}^T\underline{y}-2\underline{y}^T(\underline{\phi}+\underline{\theta})+(\underline{\phi}+\underline{\theta})^T(\underline{\phi}+\underline{\theta})\tag{14}$$

subject to the convexity constraints (2), where $\underline{\theta}$ is a n-vector such that its Euclidian norm $||\underline{\theta}||_2$ is small. For a general treatment of perturbation in nonlinear programming see Fiacco and McCormick [7]. We assume that strict complementarity holds, that is $\lambda_i^* > 0$ when $y^*[x_{i-1}, x_i, x_{i+1}] = 0$, and we state the following theorem.

Theorem 1. *Let $\underline{y}^*(\underline{\theta})$ be the solution to the perturbed problem and let $\underline{\lambda}^*(\underline{\theta})$ be the vector of the associated Lagrange multipliers. Then $\underline{y}^*(\underline{\theta})$ and $\underline{\lambda}^*(\underline{\theta})$ tend to $\underline{y}^*$ and $\underline{\lambda}^*$, respectively, as $\underline{\theta}$ tends to zero.*

Proof. Let A be the matrix whose columns are the vectors $\underline{a}_i$, $i \in \mathcal{A}^*$. The system of the vector equations (5) and (6) can be written in terms of A as

$$\underline{y}^* = \underline{\phi} + \tfrac{1}{2} A \underline{\lambda}^* \tag{15}$$

and

$$A^T \underline{y}^* = \underline{0}, \tag{16}$$

where we write A^T for the transpose of A. We multiply (15) by A^T and after taking into account (16) we solve for $\underline{\lambda}^*$ and obtain

$$\underline{\lambda}^* = -2(A^T A)^{-1} A^T \underline{\phi}. \tag{17}$$

Then we substitute (17) into (15) and obtain

$$\underline{y}^* = (I - A(A^T A)^{-1} A^T) \underline{\phi}. \tag{18}$$

We are going to prove that the solution of the perturbed problem satisfies the Karush–Kuhn–Tucker conditions for $\underline{y}^*$. Let $\underline{y}(\underline{\theta})$ and $\underline{\lambda}(\underline{\theta})$ provide the unique solution to the system of the $n + |\mathcal{A}^*|$ linear equations

$$\underline{y} - (\underline{\phi} + \underline{\theta}) = \tfrac{1}{2} A \underline{\lambda} \tag{19}$$

$$A^T \underline{y} = \underline{0}. \tag{20}$$

We solve this system for $\underline{\lambda}(\underline{\theta})$ and $\underline{y}(\underline{\theta})$ and obtain

$$\underline{\lambda}(\underline{\theta}) = -2(A^T A)^{-1} A^T (\underline{\phi} + \underline{\theta}) \tag{21}$$

and

$$\underline{y}(\underline{\theta}) = (I - A(A^T A)^{-1} A^T)(\underline{\phi} + \underline{\theta}). \tag{22}$$

Next we show that $\underline{y}(\underline{\theta})$ and $\underline{\lambda}(\underline{\theta})$ are feasible. Indeed, since, in view of (20), the equations $\underline{a}_i^T \underline{y}(\underline{\theta}) = 0,\ i \in \mathcal{A}^*$ hold, it remains to prove that $\underline{a}_i^T \underline{y}(\underline{\theta}) > 0,\ i \notin \mathcal{A}^*$. By substituting (22) on the left-hand side of the latter inequality we have

$$\underline{a}_i^T \underline{y}(\underline{\theta}) = \underline{a}_i^T (I - A(A^T A)^{-1} A^T)(\underline{\phi} + \underline{\theta})$$

$$= \underline{a}_i^T (I - A(A^T A)^{-1} A^T)\underline{\phi} + \underline{a}_i^T (I - A(A^T A)^{-1} A^T)\underline{\theta}$$

$$= \underline{a}_i^T \underline{y}^* + \underline{a}_i^T (I - A(A^T A)^{-1} A^T)\underline{\theta}. \tag{23}$$

As $\underline{a}_i^T \underline{y}^* > 0,\ i \notin \mathcal{A}^*$, it follows that for sufficiently small $||\underline{\theta}||_2$ the inequalities $\underline{a}_i^T \underline{y}(\underline{\theta}) > 0,\ i \notin \mathcal{A}^*$ are satisfied.

Further, we define $\lambda_i(\underline{\theta}) = 0$ for all integers $i \notin \mathcal{A}^*$ and we prove that $\lambda_i(\underline{\theta}) > 0,\ i \in \mathcal{A}^*$. From (21) we obtain

$$\underline{\lambda}(\underline{\theta}) = -2(A^T A)^{-1} A^T \underline{\phi} - 2(A^T A)^{-1} A^T \underline{\theta}$$

$$= \underline{\lambda}^* - 2(A^T A)^{-1} A^T \underline{\theta} \tag{24}$$

and since $\lambda_i^* > 0,\ i \in \mathcal{A}^*$, we deduce that $\lambda_i(\underline{\theta}) > 0,\ i \in \mathcal{A}^*$ for sufficiently small $||\underline{\theta}||_2$. The feasibility of $\underline{y}(\underline{\theta})$ and $\underline{\lambda}(\underline{\theta})$ and Eqs. (19) and (20) constitute the Karush–Kuhn–Tucker conditions for the perturbed problem. Thus $\underline{y}^*(\underline{\theta}) = \underline{y}(\underline{\theta})$ and $\underline{\lambda}^*(\underline{\theta}) = \underline{\lambda}^*(\underline{\theta})$. The proof of the theorem is complete. ■

The theorem states that in the absence of degeneracy (i.e. $\underline{a}_i^T \underline{y} = 0$, for some indices $i \notin \mathcal{A}^*$) small changes in the elements of $\underline{\phi}$ do not affect the optimal set of active constraints. For if $\lambda_i^* > 0,\ i \in \mathcal{A}^*$ and $\underline{a}_i^T \underline{y}^* > 0,\ i \notin \mathcal{A}^*$, they will remain so for sufficiently small changes; in addition, the row rank of A^T retains its value in the perturbed problem. However, these changes do induce some changes to $\underline{y}^*$ and $\underline{\lambda}^*$.

A corollary of Theorem 1 that provides useful information regarding the sensitivity of the convex fit is that, treating (18) as an identity in $\underline{\phi}$, differentiating and evaluating at $\underline{y}^*$ yields

$$\frac{d\underline{y}^*}{d\underline{\phi}^T} = \begin{pmatrix} \frac{\partial y_1^*}{\partial \phi_1} & \cdots & \frac{\partial y_1^*}{\partial \phi_n} \\ \vdots & & \vdots \\ \frac{\partial y_n^*}{\partial \phi_1} & \cdots & \frac{\partial y_n^*}{\partial \phi_n} \end{pmatrix} = I - A(A^T A)^{-1} A^T. \tag{25}$$

This relation shows that the effects of small finite non-zero data changes to the solution components are determined by the elements of A. Given that A has full column rank, the matrix $I - A(A^T A)^{-1} A^T$ is positive semi-definite. Hence the diagonal

elements $\partial y_i^*/\partial \phi_i$ are nonnegative and ∂y_i^* has the sign of $\partial \phi_i$. Using (22), (18) and (25) we derive the expression

$$\underline{y}(\underline{\theta}) = \underline{y}^* + \frac{d\underline{y}^*}{d\underline{\phi}^T}\underline{\theta}.$$

Analogously, treating (17) as an identity in $\underline{\phi}$, differentiating and evaluating at $\underline{\lambda}^*$ yields

$$\frac{d\underline{\lambda}^*}{d\underline{\phi}^T} = \begin{pmatrix} \frac{\partial \lambda_1^*}{\partial \phi_1} & \cdots & \frac{\partial \lambda_1^*}{\partial \phi_n} \\ \vdots & & \vdots \\ \frac{\partial \lambda_{|\mathcal{A}^*|}^*}{\partial \phi_1} & \cdots & \frac{\partial \lambda_{|\mathcal{A}^*|}^*}{\partial \phi_n} \end{pmatrix} = -2(A^T A)^{-1} A^T. \tag{26}$$

Using (21), (17) and (26) we derive the expression

$$\underline{\lambda}(\underline{\theta}) = \underline{\lambda}^* + \frac{d\underline{\lambda}^*}{d\underline{\phi}^T}\underline{\theta}.$$

We have stated the changes in the solution and the Lagrange multipliers due to small changes in the data. Further, considering specific changes of one or a few components of $\underline{\phi}$ also deserves some attention. We let $\underline{\tilde{\phi}}$ be the vector of data after we replace one or a few components by their perturbed values. If the original components are perturbed by a sufficiently small amount then, according to Theorem 1, the feasibility of the constraints $y^*[x_{i-1}, x_i, x_{i+1}] \geq 0$, for $i = 2, 3, \ldots, n-1$ is preserved. However, we assume that these components are moved so much from their original positions, that the active set is changed. Thus, another optimal vector should be calculated. We elaborate on this by assuming that the difference $y^*[x_{k-1}, x_k, x_{k+1}]$, for some $k \in [2, n-1]$ is positive and that $\underline{\phi}$ moves from its original position, so much as the kth constraint is violated by $\underline{\tilde{y}}$, say, the vector that minimizes the sum of the squares $\sum_{i=1}^{n} (\tilde{\phi}_i - y_i)^2$ subject to the equality constraints satisfied by $\underline{y}^*$, while all the other constraints remain feasible. Then the kth constraint has to be added to the active set as it is justified below [4].

Proposition 1. *We assume the conditions and we employ the notation of the previous paragraph. We assume that* $\underline{\tilde{y}}$ *gives the inequality*

$$\tilde{y}[x_{k-1}, x_k, x_{k+1}] < 0 \tag{27}$$

and that $\underline{\tilde{y}}^*$ *minimizes*

$$||\underline{\tilde{\phi}} - \underline{y}||_2^2 = \sum_{i=1}^{n} (\tilde{\phi}_i - y_i)^2$$

subject to constraints (2). Then $\underline{\tilde{y}}^*$ *satisfies the equation* $\tilde{y}^*[x_{k-1}, x_k, x_{k+1}] = 0$.

Proof. Due to the constraints satisfied by $\underline{\tilde{y}}$ and the definition of $\underline{\tilde{y}}^*$, the set of constraints satisfied by $\underline{\tilde{y}}^*$ includes the set of constraints satisfied by $\underline{\tilde{y}}$. Hence,

$$||\underline{\tilde{\phi}} - \underline{\tilde{y}}||_2 \leq ||\underline{\tilde{\phi}} - \underline{\tilde{y}}^*||_2.$$

Because, in view of (27), $\underline{\tilde{y}}$ is not feasible while $\underline{\tilde{y}}^*$ is optimal, we deduce that $\underline{\tilde{y}} \neq \underline{\tilde{y}}^*$. Therefore the former inequality is strict. Then, we assume that $\tilde{y}^*[x_{k-1}, x_k, x_{k+1}] > 0$ and we obtain a contradiction. Indeed, there exists a real number $\rho \in (0,1)$ such that the n-vector $\underline{\psi}(\rho) = \rho\underline{\tilde{y}}^* + (1-\rho)\underline{\tilde{y}}$ satisfies the constraints on $\underline{\tilde{y}}^*$. Hence

$$||\underline{\tilde{\phi}} - \underline{\psi}(\rho)||_2 < \rho||\underline{\tilde{\phi}} - \underline{\tilde{y}}^*||_2 + (1-\rho)||\underline{\tilde{\phi}} - \underline{\tilde{y}}||_2 < ||\underline{\tilde{\phi}} - \underline{\tilde{y}}^*||_2,$$

which contradicts the optimality of $\underline{\tilde{y}}^*$. It follows that $\underline{\tilde{y}}^*$ satisfies the equation $\tilde{y}^*[x_{k-1}, x_k, x_{k+1}] = 0$. ■

For perturbation analyses concerning quadratic programming and least squares one may consult, for example [1, 11, 15].

4 An Illustrative Example

We consider the following data set, in order to illustrate some points concerning the effect of changes of specific data values to the solution. Let $n = 12$ and let the data be $\{(x_i, \phi_i) : i = 1, 2, \ldots, n\}$, where x_i and ϕ_i are shown in the second and third columns of the upper left part of Table 1. Except of ϕ_1, the data are symmetric. This data set has been written by Demetriou and Powell [5] to clarify some features of the $O(n)$ initial procedure referred to in Sect. 2. In our work the small size and the symmetry of the data set make it easy to display and compare the results of the mentioned changes in short.

We applied the software package L2CXFT to $\{(x_i, \phi_i) : i = 1, 2, \ldots, 12\}$ and the solution components y_i^*, $i = 1, 2, \ldots, 12$, the Lagrange multipliers λ_i^*, $i = 2, 3, \ldots, 11$ and the associated second divided differences $\delta^2(y_i^*) = y^*[x_{i-1}, x_i, x_{i+1}]$, $i = 2, 3, \ldots, 11$ are presented in the fourth column (label y_i^*), the fifth column (label λ_i^*) and the sixth column (label $\delta^2(y_i^*)$), respectively, of the upper left part of Table 1. The sum of squares of residuals has the value $\Phi(\underline{y}^*) = \sum_{i=1}^{n} (\phi_i - y_i^*)^2 = 3.2$. Figure 1a displays the data and the solution. We see that the solution interpolates the data at x_i, for $i = 1, 3, 7$ and 11 and that $\mathcal{A}^* = \{3, 4, 5, 9, 10, 11\}$, for it consists of the integers $\{i : \delta^2(y_i^*) = 0\}$.

The nonnegativity of the sequence of the second divided differences in column 6 shows the feasibility, i.e. convexity, of the values presented in column 4. Since points with zero second divided differences lie on a straight line and since the positive second divided differences are centred at the abscissae with indices $\{2, 6, 7, 8\}$,

Table 1 The example in Sect. 4

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
						$\tilde{\phi}_7 = -5$				$\tilde{\phi}_7 = -2$				$\tilde{\phi}_7 = 1$			
i	x_i	ϕ_i	y_i^*	λ_i^*	$\delta^2(y_i^*)$	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$
1	0.999	5.405	5.405			5.405	5.695		–	5.405	5.437			5.405	5.405		
2	1.000	5.000	5.400	0	3.596	5.000	5.693	0.001	0	5.000	5.436	0	0	5.000	5.308	0	95.904
3	2.000	4.000	4.000	0.800	0	4.000	4.001	1.967	0	4.000	4.001	0.935	0	4.000	4.000	0.615	0
4	3.000	3.000	2.600	1.600	0	3.000	2.308	3.935	0	3.000	2.566	1.872	0	3.000	2.692	1.231	0
5	4.000	2.000	1.200	1.600	0	2.000	0.616	4.520	0	2.000	1.132	1.941	0	2.000	1.385	1.231	0
6	5.000	−1.000	−0.200	0	0.550	−1.000	−1.077	2.337	0	−1.000	−0.303	0.274	0	−1.000	0.077	0	0.654
7	6.000	−0.500	−0.500	0	0.300	−5.000	−2.769	0	1.706	−2.000	−1.738	0	1.436	1.000	0.077	0.923	0
8	7.000	−1.000	−0.200	0	0.550	−1.000	−1.050	2.125	0	−1.000	−0.300	0.251	0	−1.000	0.077	0	0.654
9	8.000	2.000	1.200	1.600	0	2.000	0.669	4.151	0	2.000	1.137	1.901	0	2.000	1.385	1.231	0
10	9.000	3.000	2.600	1.600	0	3.000	2.387	3.513	0	3.000	2.575	1.825	0	3.000	2.692	1.231	0
11	10.000	4.000	4.000	0.800	0	4.000	4.106	1.650	0	4.000	4.013	0.900	0	4.000	4.000	0.615	0
12	11.000	5.000	5.400			5.000	5.825			5.000	5.450			5.000	5.308		

		$\tilde{\phi}_{10} = 2.9$				$\tilde{\phi}_{10} = 5$				$\tilde{\phi}_{12} = 2.5$				$\tilde{\phi}_i = \text{rnd}$			
i	x_i	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$	$\tilde{\phi}_i$	$\tilde{y}_i^*$	$\tilde{\lambda}_i^*$	$\delta^2(\tilde{y}_i^*)$
1	0.999	5.405	5.405			5.405	5.405			5.405	5.405			4.465	4.893		
2	1.000	5.000	5.400	0	3.596	5.000	5.400	0	3.596	5.000	5.400	0	3.596	5.059	4.892	0.001	0
3	2.000	4.000	4.000	0.800	0	4.000	4.000	0.800	0	4.000	4.000	0.800	0	3.390	3.661	0.525	0
4	3.000	3.000	2.600	1.600	0	3.000	2.600	1.600	0	3.000	2.600	1.600	0	2.725	2.429	1.590	0
5	4.000	2.000	1.200	1.600	0	2.000	1.200	1.600	0	2.000	1.200	1.600	0	2.465	1.197	2.062	0
6	5.000	−1.000	−0.200	0	0.550	−1.000	−0.200	0	0.550	−1.000	−0.200	0	0.550	−1.066	−0.034	0	0.375
7	6.000	−0.500	−0.500	0	0.290	−0.500	−0.500	0	0.500	−0.500	−0.500	0	0.550	−0.516	−0.516	0	0.364
8	7.000	−1.000	−0.220	0	0.560	−1.000	0.200	0	0.350	−1.000	0.300	0	0.050	−1.879	−0.270	0	0.540
9	8.000	2.000	1.180	1.560	0	2.000	1.600	2.400	0	2.000	1.200	2.600	0	2.802	1.057	3.218	0
10	9.000	2.900	2.580	1.480	0	5.000	3.000	4.000	0	3.000	2.100	3.600	0	2.999	2.384	2.946	0
11	10.000	4.000	3.980	0.760	0	4.000	4.400	1.600	0	4.000	3.000	2.800	0	3.681	3.710	1.444	0
12	11.000	5.000	5.380			5.000	5.800			2.500	3.900			4.316	5.037		

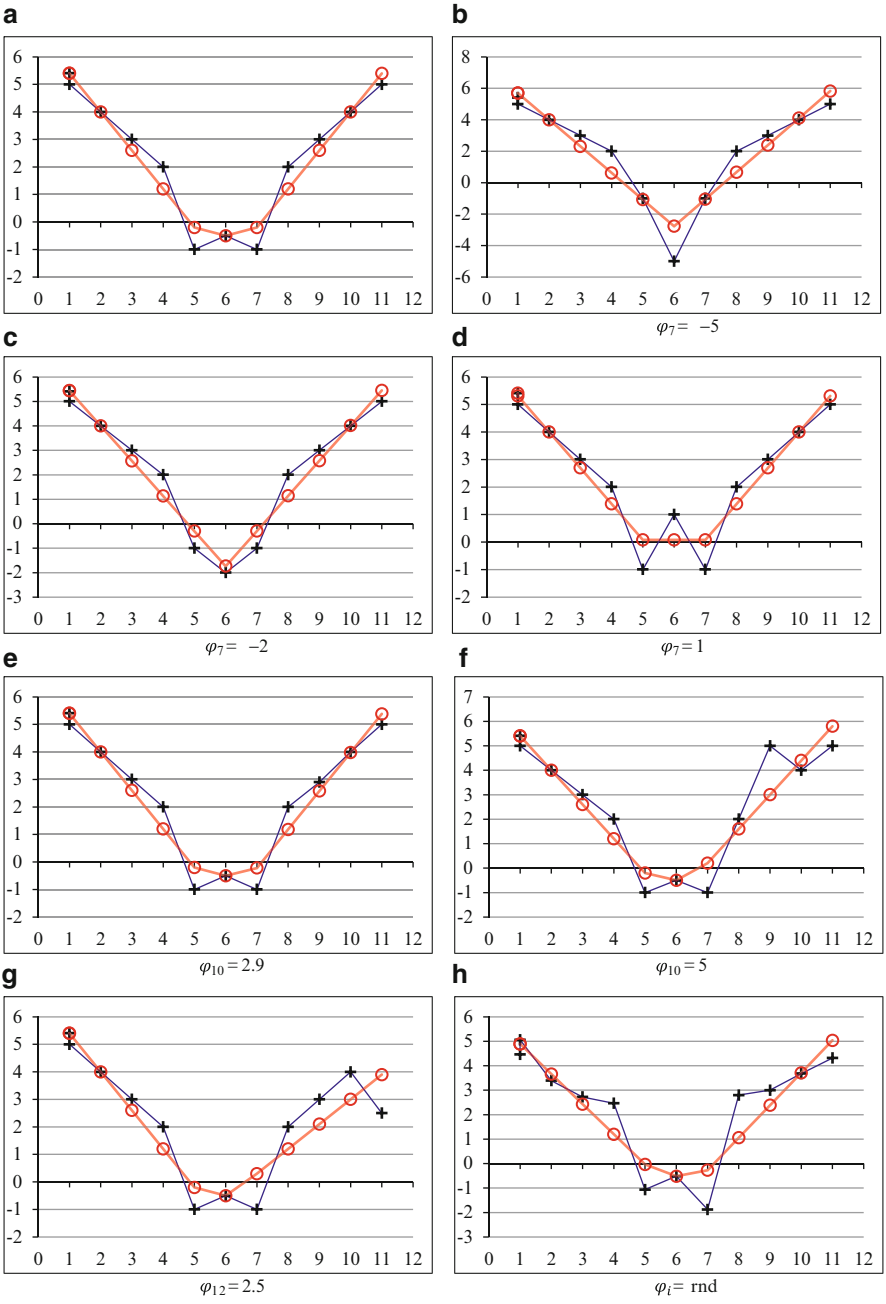


Fig. 1 Illustration of the data sets in Table 1. Convex fit (*circle*) to the data (*plus*): Figures (**a**)–(**d**) correspond to the data sets of the upper part of Table 1 and figures (**e**)–(**h**) correspond to the data sets of the lower part. *Solid lines* are for illustration

namely the knots, the calculated smoothed values lie on a convex polygonal line that consists of five consecutive line segments which join the smoothed values at the abscissae x_1, x_2, x_6, x_7, x_8 and x_{12}.

Further, it is straightforward to calculate the rates of change of the fit, i.e. the slopes of the sides of the polygon between these abscissae. Indeed, the rates are $-5, -1.4, -0.3, 0.3$ and 1.4, with respect to the intervals $[0.999, 1], [1, 5], [5, 6], [6, 7]$ and $[7, 11]$. Thus the solution components decrease in the range $[0.999, 6]$ and subsequently increase in the range $[6, 11]$, with a rate of change that increases gradually from negative to positive values.

The Lagrange multipliers in column 5 are all nonnegative and strict complementarity is immediately verified due to

$$\lambda_i^* \, y^*[x_{i-1}, x_i, x_{i+1}] = 0, \; i = 2, 3, \ldots, 11.$$

Moreover, since an active constraint corresponds to a positive Lagrange multiplier, there are no instances of degeneracy in this calculation.

We consider next how changes in the data $\underline{\phi}$ influence the components of the solution $\underline{y}^*$. In view of the 6×12 matrix A^T

$$A^T = \begin{pmatrix} 0 & 0.5 & -1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & -1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & -1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & -1 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & -1 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & -1 & 0.5 \end{pmatrix},$$

which is associated with $\mathcal{A}^*$, these results are given by formula (25), which takes the form

$$\frac{d\underline{y}^*}{d\underline{\phi}^T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0.4 & 0.2 & 0 & -0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.3 & 0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.2 & 0 & 0.2 & 0.4 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0.2 & 0 & -0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.3 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0.2 & 0.4 & 0.6 \end{pmatrix}.$$

Therefore the total effect on $\underline{y}^*$ of a change in $\underline{\phi}$ by $\underline{\theta}$, provided that $\mathcal{A}^*$ is preserved, can now be obtained as $\frac{d\underline{y}^*}{d\underline{\phi}^T}\underline{\theta}$.

A distinctive feature of this data set is that the inequalities

$$\phi[x_6, x_7, x_8] = -0.5 < 0 \text{ and } y^*[x_6, x_7, x_8] = 0.3 > 0$$

occur simultaneously. In words, a negative second divided difference at the data need not correspond to an active second divided difference at the solution. Now, if y_7^* is allowed to move subject to the feasibility of the solution, then feasibility is preserved for any value of y_7^* up to -0.2, where the point $(x_7, -0.2)$ lies on the line segment that joins the points $(x_6, y_6^* = -0.2)$ and $(x_8, y_8^* = -0.2)$.

Next, we allow ϕ_7 to move a certain distance from its value $\phi_7 = -0.5$ and see what happens in the values of the perturbed solution. To be specific, ϕ_7 is allowed to move to -5, -2 and 1 and the associated solutions are presented in the corresponding columns of Table 1 as we explain below. An immediate result is that the quadratic programming algorithm has to change some of the components of $\underline{y}^*$ in order to remove any constraint violations. In the notation of Sect. 3, let $\underline{\tilde{\phi}}$ be the vector of data after we replace the component ϕ_7 by its perturbed value $\tilde{\phi}_7$ and let $\underline{\tilde{y}}^*$ be the associated solution vector. The actual values of $\tilde{\phi}_i$ are given in the seventh column of Table 1. Some calculated results similar to those for $\underline{y}^*$ in columns 4, 5 and 6 are presented in columns 8, 9 and 10, respectively. In this manner, besides that Table 1 presents the results concerning $\underline{y}^*$, it presents also results for further data changes in two parts of rows as follows. The upper part includes three more quadruples of columns with similar results for the cases $\tilde{\phi}_7 = -5$, $\tilde{\phi}_7 = -2$ and $\tilde{\phi}_7 = 1$. The lower part of Table 1 consists also of similar quadruples of columns for the cases $\tilde{\phi}_{10} = 2.9$, $\tilde{\phi}_{10} = 5$ and $\tilde{\phi}_{12} = 2.5$ and one more quadruple where the data are random perturbations of the original ϕ_i labelled "$\tilde{\phi}_i = \text{rnd}$". The underlined numbers throughout Table 1 indicate the changed data components with respect to the data ϕ_i in column 3.

Once ϕ_7 has been moved low enough to the value $\tilde{\phi}_7 = -5$, the difference $\phi[x_6, x_7, x_8]$ became 0.4, thus positive, and the solution gave $\tilde{y}^*[x_6, x_7, x_8] = \delta^2(\tilde{y}_7^*) = 1.706 > 0$; the rest of the divided differences became equal to zero as we can see in column 10 of Table 1. The value of $\Phi(\underline{\tilde{y}}^*)$ has now been 10.78. Figure 1b displays the current data and solution. When ϕ_7 has been moved to -2, a value lower than the original $\phi_7 = -0.5$ but higher than -5, the components of the perturbed solution were nested between the components of $\underline{y}^*$ and $\underline{\tilde{y}}^*$ as it is verified by comparing the values in the relevant columns 4, 8 and 12. The value of Φ at the new solution has been 3.305. Figure 1c displays the current data and solution. When ϕ_7 has been moved to 1, the difference $\tilde{\phi}[x_6, x_7, x_8]$ became -2, but the constraint violation was removed at the solution resulting to the active constraint $\tilde{y}^*[x_6, x_7, x_8] = \delta^2(\tilde{y}_7^*) = 0$. The value of Φ has been 4.308. Figure 1d displays the current data and solution.

So far, the changes in the value of the mid-range component ϕ_7 have shown that the active set at the perturbed solution need not preserve $\mathcal{A}^*$. Hence, we consider

changes in the solution when the component $\phi_{10} = 3$, which lies at the right-half of the data range, moves to the values 2.9 (small change) and 5 (large change). Now the corresponding solutions have the values presented in the columns 3–10 of the lower part of Table 1. Figure 1e, f displays the data and the solutions. The active set of the solution associated with $\tilde{\phi}_{10} = 2.9$ preserves the active set of $\underline{y}^*$ as well as the values of the first seven components of $\underline{y}^*$. Furthermore, small changes are observed in the remaining solution components (column 4, lower part) as compared with the components of $\underline{y}^*$. Similar results are observed in the solution associated with $\tilde{\phi}_{10} = 5$, except that the changes in the solution components are more noticeable than before.

Furthermore, we show that the first and the last component of an optimal convex fit cannot be less than the corresponding data. Indeed, y_1^* cannot be less than ϕ_1, otherwise increasing it towards ϕ_1 cannot violate the first constraint, while it reduces the value of the objective function. Similarly, y_n^* cannot be less than ϕ_n. Notice also that whenever the strict inequality $y^*[x_{i-1}, x_i, x_{i+1}] > 0$ is obtained, then the inequality $y_i^* \geq \phi_i$ occurs. Assuming otherwise, we obtain a contradiction by increasing y_i^* so much as to retain feasibility and reduce the value of the objective function. Now we consider reducing the value of the extreme component ϕ_{12} from 5 (see column 2, upper part) to 2.5 (see column 11, lower part). The active set of the perturbed problem solution preserves $\mathcal{A}^*$ and gives $\tilde{y}_i^* = y_i^*$, for $i = 1, 2, \ldots, 7$, while some changes are observed in the remaining solution components (see column 12, lower part) as compared with the components of $\underline{y}^*$. Figure 1g displays the data and the solution, but notice that the knot at x_8 due to the value $\delta^2(\tilde{y}_8^*) = 0.05$ is non-visible because of low display resolution.

It will have been observed that these changes in a single ϕ_i have given changes in y_i^* of the same sign. For instance, when $\tilde{\phi}_7 = -2$, the value of ϕ_7 has been decreased by 1.5 and the value of y_7^* has been decreased by 1.238. Up to now, we have seen that even the change of a single component of the data is actually sufficient to introduce changes to all or almost all the components of the solution. When this change is small, the perturbations in the solution are usually insignificant. However, when several data values change, no matter how insignificant the changes are, the resulting influences on the solution vector spread over the data range. We present in columns 15–18 of the lower part of Table 1 the data after adding to the original ϕ_i (see column 2, upper part) a random number from the uniform distribution on $[-1.2, 1.2]$. It is worth noticing that although all the data have been changed, the solution preserved most of the active set of $\underline{y}^*$, giving three non-active constraints that are centred at x_6, x_7 and x_8. Figure 1h displays the data and the solution.

Table 1 gives the solution values as well as the Lagrange multipliers associated with the active constraints for alternative values of $\tilde{\phi}_7$, $\tilde{\phi}_{10}$, etc. The size of a Lagrange multiplier shows the sensitivity of the solution upon the associated active constraint. Thus, the larger the value of λ_i^*, the stronger the dependence of the solution to the placement of the ith constraint.

5 The Simulation Experiment

In this section some numerical results demonstrate the effects of changes in the annual Gini coefficients in the USA for the time period 1947 to 1996. The Gini coefficient is a measure of statistical dispersion intended to represent the income distribution of a nation's residents. It is the most commonly used measure of inequality. The coefficient varies between 0, which reflects complete equality, and 1 or 100, which indicates complete inequality. Our example has some importance since these coefficients are rather uncertain. Fifty data points were retrieved from the World Income Inequality Database of the US Bureau of Census 1997 and presented in the second and third columns of Table 2. We see that the macroeconomic trend of the Gini coefficient at the given period indicates convexity, a statement supported also by the convexity test of Georgiadou and Demetriou [9]. The convex fit to the Gini data is given in the column labelled y_i^* and the corresponding Lagrange multipliers are given in the column labelled λ_i^*. The convex fit is a linear spline with seven interior knots at the abscissae $x_7, x_{10}, x_{23}, x_{29}, x_{30}, x_{33}$ and x_{45}. Also the number of active constraints is $|\mathcal{A}^*| = 41$. We see that the Lagrange multipliers indexed at these knots are zero, while all the other multipliers are non-zero. A non-zero Lagrange multiplier corresponds to an active constraint. For example, it is straightforward to verify that the values $\lambda_2^* = 0.2964$ and $\lambda_7^* = 0$ are associated with the constraints $y^*[x_1, x_2, x_3] = 0$ and $y^*[x_6, x_7, x_8] = 0.1254 > 0$, respectively. The convex fit is illustrated in Fig. 2.

In the example of Sect. 4 we discuss effects of certain changes in the data, while the conclusions depend on each particular case. Our interest in this section lies on investigating how changes in the Gini coefficients influence the values of the convex fit. By means of a simulation technique, we make use of a large number of possible sets of data changes and compute the resulting influence on the solution vector for each of these sets. This in turn is likely to give an idea about the influence on the solution of the fact that the Gini data are uncertain. Formulae (22) and (21), which show that changes in any data values always give changes to the solution and the Lagrange multipliers, provided the theoretical ground for this computation.

In our experiment the changes were random perturbations of the $n = 50$ values of the Gini coefficient that are presented in the third column of Table 2. We provided the software L2CXFT with the data $\{(x_i, \phi_i) : i = 1, 2, \ldots, n\}$, where each component ϕ_i was generated by adding to the ith Gini coefficient a random number from the uniform distribution on $[-r, r]$, where we let $r = 0.1$. $M = 1{,}000$ such sets of data have been produced. Then, assuming $\underline{y}$ to be the solution, the changes were computed for each of the sets of random numbers, so that M vectors $\partial \underline{y}$ were obtained of the form $\partial \underline{y} = (\partial y_1, \partial y_2, \ldots, \partial y_n)^T$. We thus have M values for each element of the set $\{\partial y_1, \partial y_2, \ldots, \partial y_n\}$ and similarly for $\{\partial \lambda_2, \partial \lambda_3, \ldots, \partial \lambda_{n-1}\}$. The resulting changes are summarized in Table 2. We present four columns of the average and average absolute value of these elements, namely $\sum \partial y_j / M$, $\sum |\partial y_j| / M$, $\sum \partial \lambda_j / M$ and $\sum |\partial \lambda_j| / M$. In these 1,000 simulations the average of the instances of the numbers of final active constraints is 41 and the average of the instances of the number of the interior knots is 7 for 97 % of the cases.

Table 2 Average changes from $M = 1{,}000$ simulations in the convex fit (y_i^*) to the Gini coefficients in the USA 1947–1996 and in the associated Lagrange multipliers (λ_i^*), when Gini coefficients change by the addition of a random number in $[-0.1, 0.1]$

i	Year (x_i)	Gini	y_i^*	$\sum \partial y_j/M$	$\sum \lvert\partial y_j\rvert/M$	λ_i^*	$\sum \partial \lambda_j/M$	$\sum \lvert\partial \lambda_j\rvert/M$
1	1947	37.6	37.7482	−7.316E−05	7.317E−05			
2	1948	37.1	37.5393	−5.814E−05	5.814E−05	2.964E−01	4.777E−05	4.777E−05
3	1949	37.8	37.3304	−4.311E−05	4.311E−05	1.471E+00	5.076E−05	5.076E−05
4	1950	37.9	37.1214	−2.809E−05	2.809E−05	1.707E+00	1.220E−04	1.220E−04
5	1951	36.3	36.9125	−1.306E−05	1.306E−05	3.857E−01	2.499E−04	2.500E−04
6	1952	36.8	36.7036	1.961E−06	1.968E−06	2.893E−01	2.166E−04	2.168E−04
7	1953	35.9	36.4947	1.699E−05	1.700E−05	0	0	0
8	1954	37.1	36.4111	1.301E−05	1.303E−05	9.001E−01	−8.560E−05	8.560E−05
9	1955	36.3	36.3276	9.040E−06	9.057E−06	4.224E−01	−9.843E−05	9.843E−05
10	1956	35.8	36.2441	5.068E−06	5.088E-06	0	0	0
11	1957	35.1	36.1642	3.029E−06	3.050E−06	4.658E−01	4.291E−05	4.301E−05
12	1958	35.4	36.0844	9.900E−07	1.013E−06	3.060E+00	−9.545E−05	9.545E−05
13	1959	36.1	36.0045	−1.048E−06	1.048E−06	7.023E+00	−1.424E−04	1.424E−04
14	1960	36.4	35.9247	−3.087E−06	3.087E−06	1.080E+01	−6.313E−06	6.313E−06
15	1961	37.4	35.8448	−5.125E−06	5.125E−06	1.362E+01	−7.920E−06	7.920E−06
16	1962	36.2	35.7650	−7.164E−06	7.164E−06	1.333E+01	1.381E−04	1.397E−04
17	1963	36.2	35.6851	−9.203E−06	9.203E−06	1.217E+01	9.223E−05	9.354E−05
18	1964	36.1	35.6052	−1.124E−05	1.124E−05	9.980E+00	1.749E−04	1.763E−04
19	1965	35.6	35.5254	−1.328E−05	1.328E−05	6.802E+00	3.809E−04	3.821E−04
20	1966	34.9	35.4455	−1.532E−05	1.532E−05	3.475E+00	4.487E−04	4.496E−04
21	1967	34.8	35.3657	−1.736E−05	1.736E−05	1.238E+00	4.125E−04	4.132E−04
22	1968	34.8	35.2858	−1.940E−05	1.940E−05	1.334E−01	2.269E−04	2.272E−04
23	1969	34.9	35.2060	−2.138E−05	2.145E−05	0	0	0
24	1970	35.3	35.3024	−1.288E−05	1.291E−05	4.785E−01	−2.093E−04	2.094E−04

Table 2 (continued)

25	1971	35.5	35.3988	−4.371E−06	4.375E−06	9.619E−01	−2.606E−04	2.606E−04
26	1972	35.9	35.4952	4.135E−06	4.164E−06	1.243E+00	−3.149E−04	3.149E−04
27	1973	35.6	35.5917	1.264E−05	1.270E−05	7.143E−01	−2.056E−04	2.057E−04
28	1974	35.5	35.6881	2.115E−05	2.124E−05	1.690E−01	−4.008E−05	4.012E−05
29	1975	35.7	35.7845	2.965E−05	2.978E−05	0	4.275E−05	4.275E−05
30	1976	35.8	35.8834	3.585E−05	3.585E−05	0	5.100E−08	5.100E−08
31	1977	36.3	36.1333	−6.744E−06	6.790E−06	1.668E−01	7.902E−05	7.902E−05
32	1978	36.3	36.3832	−4.934E−05	4.934E−05	1.657E−04	−1.210E−07	2.100E−07
33	1979	36.5	36.6331	1.626E−05	1.635E−05	0	3.422E−06	3.422E−06
34	1980	36.5	36.9538	1.114E−05	1.121E−05	2.661E−01	1.588E−04	1.588E−04
35	1981	36.9	37.2745	6.011E−06	6.071E−06	1.440E+00	1.496E−04	1.496E−04
36	1982	38.0	37.5952	8.850E−07	9.330E−07	3.363E+00	4.344E−05	4.344E−05
37	1983	38.2	37.9159	−4.241E−06	4.241E−06	4.476E+00	1.233E−04	1.233E−04
38	1984	38.3	38.2366	−9.368E−06	9.368E−06	5.021E+00	2.189E−04	2.189E−04
39	1985	38.9	38.5573	−1.449E−05	1.449E−05	5.439E+00	1.720E−04	1.721E−04
40	1986	39.2	38.8779	−1.962E−05	1.962E−05	5.171E+00	8.872E−05	8.873E−05
41	1987	39.3	39.1986	−2.475E−05	2.475E−05	4.260E+00	3.085E−05	3.085E−05
42	1988	39.5	39.5193	−2.987E−05	2.989E−05	3.146E+00	6.054E−05	6.094E−05
43	1989	40.1	39.8400	−3.500E−05	3.503E−05	2.070E+00	2.004E−04	2.010E−04
44	1990	39.6	40.1607	−4.012E−05	4.016E−05	4.743E−01	1.881E−04	1.885E−04
45	1991	39.7	40.4814	−4.525E−05	4.530E−05	0	0	0
46	1992	40.4	40.9846	−3.687E−05	3.689E−05	1.088E+00	−1.153E−04	1.154E−04
47	1993	42.9	41.4879	−2.849E−05	2.849E−05	3.346E+00	−1.789E−04	1.789E−04
48	1994	42.6	41.9912	−2.011E−05	2.011E−05	2.780E+00	−1.773E−04	1.775E−04
49	1995	42.1	42.4944	−1.173E−05	1.173E−05	9.954E−01	−1.318E−04	1.320E−04
50	1996	42.5	42.9977	−3.344E−06	3.344E−06			

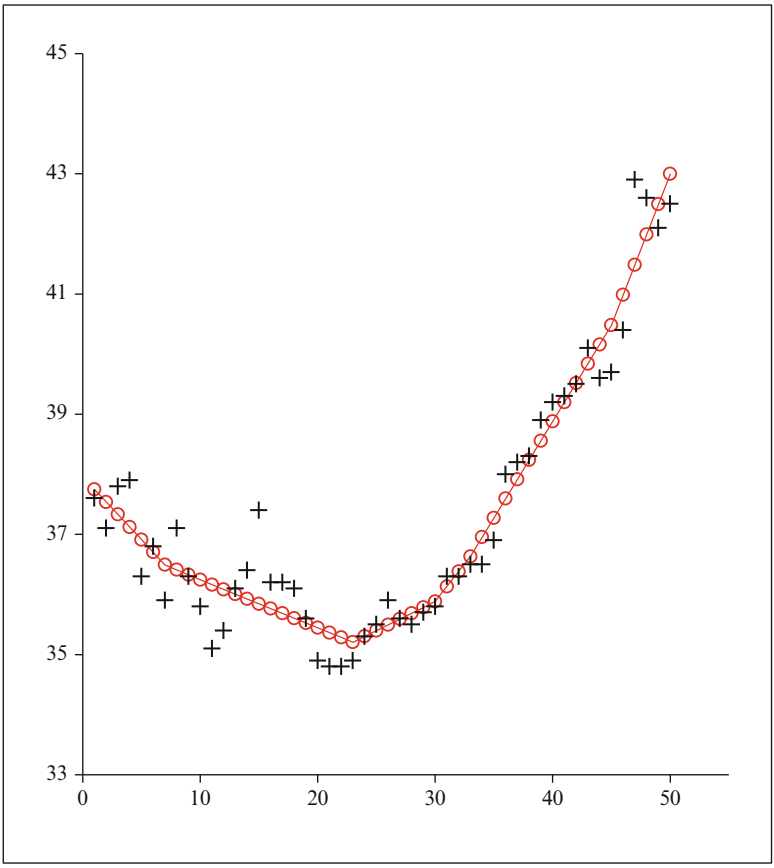


Fig. 2 Convex fit (*circle*) to the Gini coefficients (*plus*) in the USA for the time period 1947–1996

The averages of the differences between the components of the initial solution and each perturbed solution are roughly of the order of 10^{-4} and the averages of the differences between the Lagrange multipliers of the initial solution and each perturbed solution are roughly of the order of 10^{-3}. This remark is further supported by the values of the corresponding average absolute changes to the solution and to the Lagrange multipliers, which are roughly of the same order as before. These are rather minor changes indicating that the results of Sect. 3 are at least qualitatively correct. Some more experiments were tried when $r = 0.001, 0.01$, but the differences to the results are either negligible or smaller than those presented in Table 2, so we do not consider them here.

The conclusion is that the changes to the Gini coefficients considered in Table 2 made little difference to the numerical results of the initial best convex fit and the associated Lagrange multipliers. In fact, the changes in each of the considered cases either have left the initial active set unchanged or have made minor changes to it.

Indeed the knots x_7, x_{10}, x_{23} and x_{45} of the initial convex fit have been preserved by the solutions of all perturbed problems and the knots x_{30} and x_{33} have been preserved by the majority of the solutions of the perturbed problems as we deduce from the corresponding small values of the sums $\sum \partial\lambda_j/M$ and $\sum |\partial\lambda_j|/M$. Further, the check shows that the constraint centred at the abscissa 29, which has initially the small value $y^*[x_{28}, x_{29}, x_{30}] = 0.00125$, becomes active in about 3 % of the solutions of the perturbed problems, while all the resulting fits maintained the rest of the knots as already being stated. Thus the solution is robust with respect to the uncertainty of the values of the Gini coefficients at least to the accuracy considered in this section.

6 Discussion

We have been concerned with the question of sensitivity of the least squares convex fit to discrete noisy data with respect to changes in the data. The statement of the convexity in terms of nonnegative second divided differences of the data defines a strictly convex quadratic programming problem. This problem is solved very efficiently by a method that takes into account the structure of the constraints, a linear B-spline representation of the solution and the banded matrices that occur. Further, some statistical properties of the solution are studied by Hanson and Pledger [13] and Groeneboom et al. [12].

A practical question that arises in the utilization of this method is to what extent changes in the data affect the values of the solution vector. In Sect. 2 we outlined the quadratic programming method of [5] for calculating the solution that resorts upon solving positive definite tridiagonal system of equations, thus indicating stability. In Sect. 3 we stated formulae for changes in the solution and the associated Lagrange multipliers due to small changes in the data. In Sect. 4 we discussed by means of a small size illustrative example the effects of specific as well as random changes of the data to the solution. We concluded that even the change of a single component of the data is actually sufficient to introduce changes to all or almost all the components of the solution. For small changes, the perturbations in the solution are usually insignificant, but when several data values change, no matter how insignificant the changes are, the resulting influences on the solution vector spread over the data range.

As an application to real data, the sensitivity of the convex estimate to the Gini coefficient in the USA for the time period 1947–1996 was investigated in Sect. 5. The macroeconomic trend of the Gini coefficient at the given period indicates convexity. We assumed that the measurements of the Gini coefficient are considered uncertain, with a uniform probability distribution over a certain interval. The consequences of this uncertainty were investigated with a simulation technique. We computed 1,000 different sets of uniform random numbers from the interval $[-0.1, 0.1]$ that were added to the Gini measurements; these data sets were supplied to the software package L2CXFT and the changes between the solution vector of each perturbed problem and the best convex fit to the original Gini data

were calculated. The average values of these changes provided an idea about the sensitivity of the best convex fit in the original situation. In fact the average change to the solution components was of the order of 10^{-4} and the average change to the Lagrange multipliers was of the order of 10^{-3}, which shows that the convex fit is robust with respect to the uncertainty of the Gini coefficient values at least to the accuracy considered.

As a whole the results confirm that small changes to the data give only small changes to the convex fit. Furthermore, there is room for empirical analyses in utilizing the convex fit calculation as well as in investigating sensitivities, because convexity has a wide range of applications and uncertainty is everywhere in real life measurements. Our Fortran software package for the least squares convex fit would be helpful for real problem applications.

Acknowledgements This work was partially supported by the University of Athens under Research Grant 11105. The author feels grateful to the referees for valuable comments and suggestions that improved the chapter.

References

1. Boot, J.C.G.: Quadratic Programming Algorithms - Anomalies - Applications. North-Holland, Amsterdam (1964)
2. de Boor, C.: A Practical Guide to Splines, Revised Edition. Applied Mathematical Sciences, vol. 27. Springer, New York (2001)
3. Demetriou, I.C.: Algorithm 742: L2CXFT, a Fortran 77 subroutine for least squares data fitting with non-negative second divided differences. ACM Trans. Math. Softw. **21**(1), 98–110 (1995)
4. Demetriou, I.C.: Signs of divided differences yield least squares data fitting with constrained monotonicity or convexity. J. Comput. Appl. Math. **146**, 179–211 (2002)
5. Demetriou, I.C., Powell, M.J.D.: The minimum sum of squares change to univariate data that gives convexity. IMA J. Numer. Anal. **11**, 433–448 (1991)
6. Demetriou, I.C., Powell, M.J.D.: Least squares fitting to univariate data subject to restrictions on the signs of the second differences. In: Buhmann, M.D., Iserles, A. (eds.) Approximation Theory and Optimization. Tributes to M.J.D. Powell, pp. 109–132. Cambridge University Press, Cambridge (1997)
7. Fiacco, A.V., McCormick, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Wiley, London (1968)
8. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley, Chichester (2003)
9. Georgiadou, S.A., Demetriou, I.C.: A computational method for the Karush-Kuhn-Tucker test of convexity of univariate observations and certain economic applications. IAENG Int. J. Appl. Math. **38**(1), 44–53 (2008)
10. Goldfarb, D., Idnani, A.: A numerically stable dual method for solving strictly convex quadratic programs. Math. Program. **27**, 1–33 (1983)
11. Golub, G., van Loan, C.F.: Matrix Computations, 2nd edn. The John Hopkins University Press, Baltimore (1989)
12. Groeneboom, P., Jongbloed, G., Wellner, A.J.: Estimation of a convex function: characterizations and asymptotic theory. Ann. Stat. **29**, 1653–1698 (2001)
13. Hanson, D.L., Pledger, G.: Consistency in concave regression. Ann. Stat. **6**(4), 1038–1050 (1976)

14. Hildreth, C.: Point estimates of ordinates of concave functions. J. Am. Stat. Assoc. **49**, 598–619 (1954)
15. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. SIAM, Philadelphia (1995)
16. Lindley, D.V.: Making Decisions, 2nd edn. Wiley, London (1985)
17. Marchiando, J.F., Kopanski, J.J.: Regression procedure for determining the dopant profile in semiconductors from scanning capacitance microscopy data. J. Appl. Phys. **92**, 5798–5809 (2002)

Modeling and Solving Vehicle Routing Problems with Many Available Vehicle Types

Sandra Eriksson Barman, Peter Lindroth, and Ann-Brith Strömberg

Abstract Vehicle routing problems (VRPs) involving the selection of vehicles from a large set of vehicle types are hitherto not well studied in the literature. Such problems arise at Volvo Group Trucks Technology, that faces an immense set of possible vehicle configurations, of which an optimal set needs to be chosen for each specific combination of transport missions. Another property of real-world VRPs that is often neglected in the literature is that the fuel resources required to drive a vehicle along a route is highly dependent on the actual load of the vehicle.

We define the *fleet size and mix* VRP with *many* available vehicle types, called many-FSMVRP, and suggest an extended set-partitioning model of this computationally demanding combinatorial optimization problem. To solve the extended model, we have developed a method based on Benders decomposition, the subproblems of which are solved using column generation, and the column generation subproblems being solved using dynamic programming; the method is implemented with a so-called projection-of-routes procedure. The resulting method is compared with a column generation approach for the standard set-partitioning model. Our method for the extended model performs on par with column generation applied to the standard model for instances such that the two models are equivalent. In addition, the utility of the extended model for instances with very many available vehicle types is demonstrated. Our method is also shown to efficiently handle cases in which the costs are dependent on the load of the vehicle.

Computational tests on a set of extended standard test instances show that our method, based on Benders' algorithm, is able to determine combinations of vehicles and routes that are optimal to a relaxation (w.r.t. the route decision variables) of the extended model. Our exact implementation of Benders' algorithm appears, however,

S. Eriksson Barman • A.-B. Strömberg (✉)
Department of Mathematical Sciences, Chalmers University of Technology
and University of Gothenburg, Gothenburg, Sweden
e-mail: barmane@chalmers.se; anstr@chalmers.se

P. Lindroth
Chassis & Vehicle Dynamics, Volvo Group Trucks Technology, Gothenburg, Sweden
e-mail: peter.lindroth@volvo.com

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_6

too slow when the number of customers grows. To improve its performance, we suggest that relaxed versions of the column generation subproblems are solved and that the set-partitioning model is replaced by a set-covering model.

Keywords Vehicle routing problem • Fleet size and mix • Heterogeneous fleet • Many vehicle types • Set partitioning • Benders decomposition • Projection-of-routes

1 Introduction

Vehicle routing problems (VRPs) have been studied for many years. The first model and algorithm were proposed in 1959 by Dantzig [9], and since then hundreds of models and algorithms have been developed; see [25]. The VRP is a computationally demanding combinatorial optimization problem with applications in many fields, including transportation, logistics, communication, and manufacturing; they are among the most studied combinatorial optimization problems. The classical version, the *capacitated* VRP (CVRP), consists of the search for a solution to a simplified transport mission in which customers are serviced by a set of identical vehicles delivering goods from a central depot, and in which the configuration of customers that each vehicle can service on one route obeys a capacity restriction. The CVRP is an extension of the traveling salesperson problem (TSP) and is NP-hard in the strong sense; see [28, p. 8]. Nevertheless, there is a wide range of extensions of the CVRP. Research on the VRP has been successful and proved relevant in industrial applications. There is also a growing industry of software for transportation planning based on methods developed by the scientific community for the VRP, and increasingly complex models and larger sized problems are being solved [24, 28].

When modeling this type of real-world combinatorial optimization problems, decisions have to be made about what aspects should be included. A heterogenous fleet of vehicles is one aspect of real-life problems that may be important to consider. Hoff et al. [17, p. 2043] state that "there is generally a strong dependency between fleet composition and routing"; therefore the corresponding decisions need to be integrated. Given a transport mission, the optimal routing solution depends strongly on the characteristics of the available fleet of vehicles. For a company such as the Volvo Group Trucks Technology (Volvo GTT), that faces an immense set of possible vehicle configurations, the inclusion of a very large set of vehicle types is of great interest. Volvo GTT wishes to determine an optimal set of vehicle configurations for any specific combination of transport missions. This will help their customers to make more qualified vehicle purchasing decisions, which in turn will make them more satisfied. This type of optimization tool can also help Volvo GTT to better understand their customers' needs, which may then influence strategic vehicle development decisions and make Volvo GTT even more competitive on the tough global vehicle market.

Most successful heuristic methods for the VRP combine different classical heuristics. Recently combinations of exact methods—based on mathematical programming—and meta-heuristics were proposed. According to Drexl [11, p. 61], "[a]n exact solution of real-world problems with many additional side constraints will remain impossible in the short and medium term. However, close-to-optimal solutions of more and more complex and integrated problems, increasingly based on incomplete optimization approaches and mathematical-programming-based heuristics, are possible, and this is sufficient to provide useful decision support in practice".

This work focuses on developing models and algorithms appropriate for VRPs with a very large set of vehicle types. We have taken as a starting point the so-called *fleet size and mix* VRP (FSMVRP), in which a heterogeneous fleet of vehicles is available. Standard models and algorithms for the VRP with a heterogeneous fleet consider only *a few* vehicle types. We consider a *large set of vehicle types* and denote the corresponding problem many-FSMVRP. We have adapted a mathematical optimization model and an algorithm based on *column generation*, which has proved to be a successful heuristic for the FSMVRP, to accommodate a very large set of vehicle types. In addition a new model, in which the number of vehicle types that are allowed in a feasible solution is limited, and an algorithm, based on Benders decomposition, are proposed. The limit on the number of vehicles allowed, in addition to being a relevant part of the model, proves technically useful in the decomposition algorithm, for cases when the set of possible vehicle types is very large. Load-dependent costs, which are not included in standard models for the FSMVRP, are also developed and implemented, in order to illustrate how the solution framework developed can be extended to include additional properties of real transportation problems.

This chapter is structured as follows. In Sect. 2 we review relevant scientific literature. The mathematical models and algorithms developed are presented in Sects. 3 and 4, respectively. In Sect. 5 we present computational tests and results while conclusions are drawn in Sect. 6.

2 Literature Review of the VRP with a Heterogeneous Fleet

The VRP with a set of *non-identical* vehicle types was first formulated in 1984 by Golden [16]. When the number of vehicles of each type is constrained, the problem is generally known as the *heterogeneous* VRP (HVRP), otherwise it is known as the FSMVRP. The objective function may include a fixed cost for each vehicle that is used, and/or routing costs dependent on the length of each route and possibly also on the vehicle type; see [2].

Algorithms for the HVRP are often tested on the twelve instances introduced in [16] (denoted G12 in [2]) and on eight of those instances adapted by Taillard [23] to include vehicle-dependent variable costs (denoted T8 in [2]); see also [28]. However, none of the test instances in G12 and T8 possesses a combination of fixed

and variable costs that both depend on the vehicle type. Choi and Tcha [6] combined the instances G12 and T8 to a set of twelve test instances, here denoted CT12, in each of which both the fixed and the variable costs depend on the vehicle type. The instances in G12, T8, and CT12 contain three to six vehicle types each.

Until recently, no exact algorithm had been implemented for the HVRP "[d]ue to the intrinsic difficulty of this family of routing problems" [2, p. 13]. Exact solution methods have now been developed based on branch-and-cut-and-price in [21] and on a set-partitioning formulation using additional constraints in [1], the latter being able to solve instances with up to 75 customers, and some instances with 100 customers; it works well for other variants of VRPs as well (see [28]).

Hoff et al. [17] and Baldacci et al. [2] review numerous heuristic methods applied to the HVRP; both report that the column generation-based heuristic of Choi and Tcha in [6], which employs a set-covering formulation, provides good results as compared with other heuristic methods. In that heuristic, each vehicle type determines one column generation subproblem, which in turn is relaxed—both by using a state-space relaxation and by relaxing the elementary constraint of the routes—and solved using dynamic programming with a 2-cycle elimination procedure. In [17, p. 2048] it is stated that "[t]he results confirm the dominance of this algorithm, both in terms of quality and computation time". When applying the column generation algorithm in [6] to the instances in G12, solutions were found, for which the average relative gap with respect to the best known solution values at the time was 0.004 %; for T8 the best known solutions were found; see [2, Tables 4 and 6]. Optimal solutions for all but one of the instances in CT12 are presented in [1, Table 7]; the average of the relative differences between the values of each of these optimal solutions and the corresponding solution in [6] is 0.09 %.

3 Mathematical Models for the Many-FSMVRP

Algorithms for the FSMVRP presented in the literature have hitherto been focused on instances with few vehicle types, and to the standard test instances for the VRP with a homogeneous fleet contain relatively few vehicle types. An efficient handling of large instances of the FSMVRP requires models and solution techniques tailored for a large set of vehicle types. The many-FSMVRP developed and studied in this chapter is an extension of the FSMVRP adapted to the case when the number of available vehicle types is very large.

Based on the solution technique for the HVRP presented in [6], and a (simple) set-partitioning model, we have developed and implemented a *column generation* algorithm for the many-FSMVRP. We have also developed an extended set-partitioning model, which restricts the number of vehicle types that may be used in a solution to the many-FSMVRP, due to two main reasons: (a) A limit on the number of vehicle types allowed is a natural property for practical routing problems, e.g., since a fleet with fewer vehicle types can be more flexible. (b) By restricting the number of vehicle types used in a solution, a Benders decomposition of a relaxed

model is enabled, the resulting Benders subproblem being essentially equivalent to the simple set-partitioning model, although with fewer vehicle types. The number of vehicle types allowed in the extended set-partitioning model determines the number of vehicle types considered in Benders subproblem, influencing its computational complexity.

To the best of our knowledge, a limit on the number of vehicle types used in a solution to the extended set-partitioning model has not previously been implemented for the HVRP. To illustrate how our models adapt to more complex problem settings they have also been modified to consider load-dependent costs.

The proposed models of the many-FSMVRP are defined on a directed graph $(\mathcal{N},\mathcal{A})$, where $\mathcal{N} := \{0\} \cup \mathcal{N}_0$, $\mathcal{N}_0 := \{1,\ldots,N\}$ denotes the set of nodes representing the customers, node 0 represents the depot, and $\mathcal{A}$ denotes the set of directed links between pairs of nodes in $\mathcal{N}$. Each customer $i \in \mathcal{N}_0$ has a demand $d_i > 0$. The set $\mathcal{K} := \{1,\ldots,K\}$ represents the vehicle types and each vehicle type $k \in \mathcal{K}$ has a limited capacity $D_k > 0$.

Associated with each vehicle type $k \in \mathcal{K}$ are a variable cost, c^{link}_{ijk}, for each link $(i,j) \in \mathcal{A}$, and a fixed cost f_k. The variable costs are modeled as $c^{\text{link}}_{ijk} := \alpha_k \,\text{dist}(i,j)$, where $\text{dist}(i,j)$ denotes the length of link $(i,j) \in \mathcal{A}$ and the coefficient α_k increases with an increasing capacity D_k, $k \in \mathcal{K}$. For each vehicle type $k \in \mathcal{K}$ the feasible routes are implicitly defined by the index set $\mathcal{R}_k$. A route is a sequence of nodes $(i_0,i_1,\ldots,i_{H-1},i_H)$, such that $(i_{h-1},i_h) \in \mathcal{A}$, $h = 1,\ldots,H$. A route $r := (i_0,i_1,\ldots,i_{H-1},i_H)$ is feasible if it starts at the depot, visits each customer at most once, and ends at the depot, i.e., $H \geq 2$, $i_0 = i_H = 0$, $\{i_1,\ldots,i_{H-1}\} \subseteq \mathcal{N}_0$, and $i_{h_1} \neq i_{h_2}$ whenever $h_1,h_2 \in \{1,\ldots,H-1\}$ and $h_1 \neq h_2$. A route–vehicle pair (r,k) is feasible if route $r \in \mathcal{R}_k$ is feasible and does not exceed the capacity constraints of vehicle type $k \in \mathcal{K}$, i.e., if $\sum_{h=1}^{H-1} d_{i_h} \leq D_k$. The cost of a feasible route–vehicle pair (r,k) is defined as $c_{rk} := f_k + \sum_{h=1}^{H} c^{\text{link}}_{i_{h-1}i_hk}$.

3.1 *A Set-Partitioning Formulation of the* FSMVRP

The FSMVRP is to minimize the sum of the costs of the routes traveled by the vehicles, while each customer is visited by exactly one vehicle. Defining the parameters

$$\delta_{irk} := \begin{cases} 1, & \text{if the route } r\text{, of the route-vehicle} \\ & \text{pair } (r,k)\text{, visits customer } i\text{,} \\ 0, & \text{otherwise,} \end{cases} \qquad i \in \mathcal{N}_0,\ r \in \mathcal{R}_k,\ k \in \mathcal{K},$$

and the decision variables

$$x_{rk} := \begin{cases} 1, & \text{if the route-vehicle pair } (r,k) \text{ is used,} \\ 0, & \text{otherwise,} \end{cases} \qquad r \in \mathcal{R}_k, \quad k \in \mathcal{K},$$

the set-partitioning formulation of the FSMVRP is given by

$$z^* := \min_{\mathbf{x}} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} c_{rk} x_{rk}, \tag{1a}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} \delta_{irk} x_{rk} = 1, \qquad i \in \mathcal{N}_0, \tag{1b}$$

$$x_{rk} \in \{0,1\}, \qquad r \in \mathcal{R}_k, \quad k \in \mathcal{K}. \tag{1c}$$

The objective (1a) is to minimize the sum of the costs over the routes that are used. The constraints (1b) ensure that each customer is visited by exactly one vehicle. This type of set-partitioning formulation of the VRP was originally proposed in [3]. The model is general and has the advantage that many restrictions can easily be incorporated, since the feasibility of routes is implicitly defined by the sets $\mathcal{R}_k$, $k \in \mathcal{K}$. The linear programming (LP) relaxation of this type of model for the VRP is often very tight; see [24, p. 22]. Since, for realistic problem instances, the number, $\sum_{k \in \mathcal{K}} |\mathcal{R}_k|$, of feasible routes is extremely large (tens of customers may yield billions of feasible routes), column generation, possibly combined with a branch-and-bound algorithm or cut generation, is an appropriate solution method.

3.2 *An Extended Set-Partitioning Model of the* many-FSMVRP

Considering the huge set $\mathcal{K}$ of vehicle types in the many-FSMVRP, we propose an extended set-partitioning formulation, in which the number of vehicle types that may be used is limited. We define the parameters $C =$ the maximum allowed number of vehicle types (the vehicle type limit) and $M =$ the maximum allowed number of vehicles of each type, and the variables

$$y_k := \begin{cases} 1, & \text{if vehicle type } k \text{ is allowed,} \\ 0, & \text{otherwise,} \end{cases} \qquad k \in \mathcal{K}.$$

The extended set-partitioning formulation is then given by

$$z^*_{\text{EXT}} := \min_{\mathbf{x},\mathbf{y}} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} c_{rk} x_{rk}, \tag{2a}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} \delta_{irk} x_{rk} = 1, \qquad i \in \mathcal{N}_0, \tag{2b}$$

$$\sum_{r \in \mathcal{R}_k} x_{rk} \leq M y_k, \qquad k \in \mathcal{K}, \tag{2c}$$

$$\sum_{k \in \mathcal{K}} y_k \leq C, \tag{2d}$$

$$x_{rk} \in \{0,1\}, \qquad r \in \mathcal{R}_k, \quad k \in \mathcal{K}, \tag{2e}$$

$$y_k \in \{0,1\}, \qquad k \in \mathcal{K}. \tag{2f}$$

This extends the formulation (1) by the constraints (2c)–(2d) and the binary variables y_k, $k \in \mathcal{K}$. The constraints (2c) limit the number of vehicles of each type, and the constraint (2d) sets the vehicle type limit. Hence, the inequality $z^*_{\text{EXT}} \geq z^*$ holds.

3.3 Load-dependent Costs

Most VRP settings restrict the variable travel costs to depend on the distance traveled only. Since travel costs depend on many factors, this may be too limiting for a practical application. According to [29], factors determining real travel costs can be divided into two groups, of which the first includes distance, but also speed, load, fuel consumption, and road conditions. The factors in the second group are less related to the route traveled and include vehicle depreciation and maintenance costs, wages, and taxes. Most of the factors in the first group are related to fuel consumption, which is highly dependent on the distance traveled and on the vehicle load. The authors argue, using statistical data, that fuel consumption can be modeled as an affine function involving the fuel consumption rates of a fully loaded and an empty vehicle. We define the parameters $a =$ the fuel unit cost and $\rho_k^{D_k}(\rho_k^0) =$ the fuel consumption rate of a fully loaded (empty) vehicle of type $k \in \mathcal{K}$, and assume that the customer demands d_i, $i \in \mathcal{N}_0$, are given in weight units. The cost of the route–vehicle pair (r,k), where $r = (0, i_1, \ldots, i_{H-1}, 0)$, is then defined as

$$c_{rk}^{\text{load}} := f_k + a \sum_{h=1}^{H-1} \text{dist}(i_{h-1}, i_h) \left[\rho_k^0 + \frac{\rho_k^{D_k} - \rho_k^0}{D_k} \left(\sum_{t=h}^{H-1} d_{i_t} \right) \right] + a \, \text{dist}(i_{H-1}, 0) \, \rho_k^0,$$

where $\sum_{t=h}^{H-1} d_{i_t}$ represents the load of the vehicle after its visit at customer i_{h-1} along route r. Hence, the cost of the route–vehicle pair (r,k) is given by the sum of the fixed cost f_k and a weighted sum of the distances $\text{dist}(i_{h-1}, i_h)$, $h = 1, \ldots, H$, where the weights increase with the vehicle load on the links.

To adapt the modeling of load-dependent costs to our test instances we define the weight parameters $Q^{\text{dist}} > 0$ and $Q^{\text{load}} > 0$ (see Sect. 5.3 for the derivation of these values) and define the load-dependent cost of the route–vehicle pair (r,k) as

$$c_{rk}^{\text{load}} := f_k + \sum_{h=1}^{H-1} c_{i_{h-1}i_h k}^{\text{link}} \left[Q^{\text{dist}} + Q^{\text{load}} \left(\sum_{t=h}^{H-1} d_{i_t} \right) \right] + c_{i_{H-1}0k}^{\text{link}} Q^{\text{dist}}, \tag{3}$$

where the link costs c_{ijk}^{link}, $(i,j) \in \mathcal{A}$, correspond to the load-independent costs.

4 Algorithms for the many-FSMVRP

We use a column generation algorithm to solve the model (1) and a combined Benders decomposition and column generation algorithm to solve the extended model (2). The load-dependent costs are implemented by replacing c_{rk} by c_{rk}^{load} in both models and altering the column generation subproblems accordingly. For the extended model, this has implications for the Benders subproblems. The property of the set-partitioning model—on which both of our models are based—that only the subproblems need to be altered is quite useful, as will be demonstrated.

4.1 Column Generation Applied to the Set-Partitioning Model

For combinatorial optimization problems that can be formulated as set-partitioning problems with binary variables, e.g., the VRP and the crew pairing assignment problem, column generation has shown to be a successful solution strategy. Column generation is often implemented in a branch-and-bound algorithm, then called *branch-and-price*, and in which columns are generated in each node of the branch-and-bound tree; see [26]. The most successful exact algorithms for the VRP are based on branch-and-price with additional cut generation, the so-called *branch-and-cut-and-price*; see [11].

Column generation can also be used to find good, but not necessarily optimal, solutions, as a mathematical programming-based heuristic; see [27, pp. 352–353]. The algorithm implemented here for the model (1) is based on the column generation heuristic of Choi and Tcha in [6], which has proved successful for the HVRP. For a thorough account of column generation, see [26].

To apply the column generation principle, the binary requirements (1c) on the variables x_{rk} are relaxed according to

$$x_{rk} \geq 0, \qquad r \in \mathcal{R}_k, \quad k \in \mathcal{K}. \tag{4}$$

We formulate the *column generation restricted master problem* as

$$\begin{aligned}
\text{[CGRMP]} \qquad \tilde{z}_{\text{CGRMP}} := \min_{\mathbf{x}} \quad & \sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} c_{rk} x_{rk}, \\
\text{s.t.} \quad & \sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \delta_{irk} x_{rk} = 1, \qquad i \in \mathcal{N}_0, \\
& x_{rk} \geq 0, \qquad r \in \tilde{\mathcal{R}}_k, \quad k \in \mathcal{K},
\end{aligned}$$

where $\tilde{\mathcal{R}}_k \subseteq \mathcal{R}_k$, $k \in \mathcal{K}$. Note that, when $\tilde{\mathcal{R}}_k = \mathcal{R}_k$ it holds that [CGRMP] is equivalent to (1a)–(1b), (4). The LP dual of [CGRMP] is formulated as

$$\begin{aligned}
\text{[CGRMPDual]} \qquad \tilde{z}_{\text{CGRMP}} = \max_{\boldsymbol{\pi}} \quad & \sum_{i\in\mathcal{N}_0} \pi_i, \\
\text{s.t.} \quad & \sum_{i\in\mathcal{N}_0} \delta_{irk} \pi_i \leq c_{rk}, \qquad r \in \tilde{\mathcal{R}}_k, \quad k \in \mathcal{K}.
\end{aligned}$$

The sets $\tilde{\mathcal{R}}_k$ are initialized such that there exists a feasible solution to [CGRMP]. In each iteration of the column generation algorithm, the sets $\tilde{\mathcal{R}}_k$ are expanded by routes $r \in \mathcal{R}_k \setminus \tilde{\mathcal{R}}_k$, possessing low (negative) reduced costs, until an optimal solution to the model (1a)–(1b), (4) is found and verified (i.e., until all reduced costs are nonnegative). We denote the corresponding optimal value by z^*_{LP}. Finally, to obtain a feasible solution to (1), binary restrictions on the variables x_{rk} are added to [CGRMP], which is solved to optimality with respect to the columns generated.

4.1.1 Adding Routes to the Restricted Master Problem

For $r \in \mathcal{R}_k$, $k \in \mathcal{K}$, the reduced cost of the variable x_{rk}, denoted $\hat{c}_{rk}$, is given by

$$\hat{c}_{rk} := c_{rk} - \sum_{i\in\mathcal{N}_0} \delta_{irk} \pi^*_i, \tag{6}$$

where $\boldsymbol{\pi}^*$ denotes an optimal solution to [CGRMPDual]. Defining

$$\hat{c}^{\text{link}}_{ijk} := c^{\text{link}}_{ijk} - \pi^*_j, \qquad (i,j) \in \mathcal{A}, \quad j \in \mathcal{N}_0, \tag{7a}$$

$$\hat{c}^{\text{link}}_{i0k} := c^{\text{link}}_{i0k}, \qquad (i,0) \in \mathcal{A}. \tag{7b}$$

the reduced cost $\hat{c}_{rk}$ can be expressed as

$$\hat{c}_{rk} = f_k + \sum_{h=1}^{H} c^{\text{link}}_{i_{h-1} i_h k} - \sum_{h=1}^{H-1} \pi^*_{i_h} = f_k + \sum_{h=1}^{H} \hat{c}^{\text{link}}_{i_{h-1} i_h k}.$$

Following [6], define one column generation subproblem per vehicle type $k \in \mathcal{K}$ as

$$\hat{c}_k^* := \min_{r \in \mathcal{R}_k} \{\hat{c}_{rk}\} = \min_{r \in \mathcal{R}_k} \left\{ c_{rk} - \sum_{i \in \mathcal{N}_0} \delta_{irk} \pi_i^* \right\} = f_k + \min_{r \in \mathcal{R}_k} \left\{ \sum_{h=1}^{H} \hat{c}^{\text{link}}_{i_{h-1} i_h k} \right\}. \tag{8}$$

Each problem (8) is an *elementary shortest path problem with resource constraints*, denoted ESPPRC in [18], which is NP-hard in the strong sense. This is related to the fact that the network may contain at least one cycle, such that the sum over its links of the reduced link costs (7) is negative; see [18, 22]. Dynamic programming is commonly used for solving the ESPPRC subproblems when column generation is used to solve the VRP. Due to the complexity of the ESPPRC, often a (relaxed) non-elementary *shortest path problem with resource constraints* (SPPRC), which possesses a pseudo-polynomial complexity, is solved; see [14, 22]. We solve the ESPPRC using dynamic programming—following [14]—in the first part of the column generation algorithm. As a means to improve the computational complexity of the algorithm, we suggest that these subproblems are relaxed into SPPRCs.

The following measures were found to provide substantial savings in computation time. In order to speed up the convergence towards the optimal value of the model (1a)–(1b), (4), the column generation subproblems have been solved approximately (except in the last iteration, when verifying that the solution to [CGRMP] is optimal in the relaxed model); see [20]. Hence, a route $r \in \mathcal{R}_k \setminus \tilde{\mathcal{R}}_k$, such that the variable x_{rk} possesses a negative but not necessarily minimal reduced cost $\hat{c}_{rk} = c_{rk} - \sum_{i \in \mathcal{N}_0} \delta_{irk} \pi_i^*$, has been added to $\tilde{\mathcal{R}}_k$. In addition, since the original subproblem—finding a variable x_{rk} with minimal reduced cost—is divided into one subproblem for each vehicle type k, the subproblem for each vehicle type does not have to be solved in every iteration. Instead a so-called *partial column generation* has been employed in which only a subset of the subproblems is considered in each iteration. However, at least one variable x_{rk} with a negative reduced cost must be added to [CGRMP], provided that such a variable exists; see [20]. Since there are as many subproblems as vehicle types, the set of which is assumed to be very large, the use of partial column generation has shown to be beneficial.

We have used two different approaches to solve the subproblems (8), one based on a mathematical model solved by AMPL and CPLEX, and the other based on dynamic programming implemented in Matlab. Both approaches have been implemented such that the solution algorithm can be interrupted before an optimal solution to the subproblem has been found, either when a predefined time limit has been exceeded or after a certain number of routes with negative reduced cost have been found. We noted that often when no route with a negative reduced cost had been found for a given subproblem in the later column generation iterations, then no such route was found in the next couple of iterations either. Therefore, a kind of *tabu strategy of partial column generation* was also implemented for the subproblems (see [15] for an introduction to tabu search) according to the following. If, for one specific subproblem, no route with a negative reduced cost has been found during a predetermined number of consecutive column generation iterations,

this subproblem is not solved for a predetermined number of iterations. We also noted that subproblems corresponding to vehicles with similar capacities sometimes yielded the same routes—both when solved to optimality and when not. The tabu strategy was then updated, so that only the route–vehicle pair possessing the lowest reduced cost was added to the restricted master problem, and the other subproblems (which yielded the same route) were recorded as *not* providing a route with negative reduced cost—thus potentially leading to these not being solved for a number of iterations. For more details about the implementation of the subproblem solver, see [12].

4.1.2 Terminating the Column Generation Algorithm

As previously mentioned, the optimal solution to the final [CGRMP] is an optimal solution to the model (1a)–(1b), (4), when the reduced costs of all the feasible route–vehicle pairs are nonnegative. However, in the later iterations of the column generation often only very small improvements are made; the so-called tailing off effect; see [20]. Hence, terminating the algorithm prior to convergence may be beneficial.

In each iteration of the column generation algorithm, the optimal solution to [CGRMP] is feasible in the model (1a)–(1b), (4), which possesses the optimal value z^*_{LP}. Hence, $\tilde{z}_{\text{CGRMP}}$ is and upper bound on z^*_{LP}. A lower bound on z^*_{LP} is given by $\underline{z} := N \cdot \min_{k \in \mathcal{K}} \{\hat{c}^*_k\} + \tilde{z}_{\text{CGRMP}}$, where $\hat{c}^*_k$ is found by solving the subproblem in (8) to optimality (see [12, Sect. 4.1.2] for a derivation of $\underline{z}$). These upper and lower bounds on z^*_{LP} have been used to terminate the column generation prior to convergence, i.e., when $\tilde{z}_{\text{CGRMP}} - \underline{z} \leq \varepsilon$, for some predetermined value of $\varepsilon > 0$.

4.2 Benders' Decomposition Algorithm for the Extended Set-Partitioning Model

To handle the large set of vehicle types in the many-FSMVRP, decomposing the problem in several levels has shown fruitful. For routing and scheduling problems arising in airline planning, Benders decomposition combined with column generation was successfully applied by Cordeau et al. in [8] when considering the simultaneous aircraft routing and crew pairing—due to the high complexity of the problem this has traditionally been done in sequence. In a Benders decomposition of an optimization problem, in each iteration one set of variables—called *complicating variables*—is fixed, and the (restricted) problem with respect to the remaining variables is solved to optimality. This is iterated: in each iteration, information about solutions from former iterations is used to fix the complicating variables, until an optimal solution to the restricted problem is verified to be optimal in the original problem. See [19] for an account of Benders decomposition.

One advantage of Benders decomposition compared to column generation is that the former procedure can handle binary requirements on the variables. For the many-FSMVRP, this allows for keeping the binary requirements on the variables y_k in the extended set-partitioning model, resulting in Benders subproblems in which only subsets of the vehicle types are allowed. These subproblems can then be solved by the column generation algorithm described in Sect. 4.1.

First, to use Benders' algorithm, we relax the binary requirements (2e) on the variables x_{rk}, according to (4). The optimal value of the resulting model (2a)–(2d), (2f), (4) is denoted v^*. We consider the variables y_k, $k \in \mathcal{K}$, as complicating and define the set $\mathcal{K}^{\text{cap}} := \left\{ k \in \mathcal{K} \,\middle|\, D_k \geq \max_{i \in \mathcal{N}_0} \{d_i\} \right\}$ of the vehicle types possessing a capacity large enough to service any of the customers. The set

$$S := \left\{ \mathbf{y} \in \{0,1\}^K \,\middle|\, \sum_{k \in \mathcal{K}} y_k \leq C, \sum_{k \in \mathcal{K}^{\text{cap}}} y_k \geq 1 \right\}$$

contains the values of the complicating variables $\mathbf{y}$ for which the remaining problem in the variables $\mathbf{x}$—i.e., *the Benders subproblem*—possesses at least one feasible solution. For fixed values of the variables $\mathbf{y} := \tilde{\mathbf{y}}$ this subproblem appears as

$$\begin{aligned}
[\text{BendersSP}(\tilde{\mathbf{y}})] \qquad w^*(\tilde{\mathbf{y}}) := \min_{\mathbf{x}} \quad & \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} c_{rk} x_{rk}, \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} \delta_{irk} x_{rk} = 1, \qquad i \in \mathcal{N}_0, \\
& \sum_{r \in \mathcal{R}_k} x_{rk} \leq M \tilde{y}_k, \quad k \in \mathcal{K}, \\
& x_{rk} \geq 0, \qquad r \in \mathcal{R}_k, \quad k \in \mathcal{K},
\end{aligned}$$

and its corresponding LP dual problem is given by

$$[\text{BendersSPDual}(\tilde{\mathbf{y}})] \quad w^*(\tilde{\mathbf{y}}) = \max_{\boldsymbol{\pi}, \boldsymbol{\gamma}} \quad \left[\sum_{i \in \mathcal{N}_0} \pi_i + M \sum_{k \in \mathcal{K}} \tilde{y}_k \gamma_k \right], \tag{9a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{N}_0} \delta_{irk} \pi_i + \gamma_k \leq c_{rk}, \quad r \in \mathcal{R}_k, \quad k \in \mathcal{K}, \tag{9b}$$

$$\gamma_k \leq 0, \quad k \in \mathcal{K}. \tag{9c}$$

We denote the dual feasible set by

$$F_{\text{BendersSPDual}} := \left\{ (\boldsymbol{\pi}, \boldsymbol{\gamma}) \in \mathbb{R}^{N+K} \,\middle|\, (\boldsymbol{\pi}, \boldsymbol{\gamma}) \text{ satisfies (9b)–(9c)} \right\}.$$

Defining a constrained set of vehicles as $\tilde{\mathcal{K}}(\mathbf{y}) := \{k \in \mathcal{K} \,|\, y_k = 1\}$, an equivalent formulation to [BendersSP($\tilde{\mathbf{y}}$)] is given by

$$w^*(\tilde{\mathbf{y}}) = \min_{\mathbf{x}} \sum_{k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}})} \sum_{r \in \mathcal{R}_k} c_{rk} x_{rk}, \tag{10a}$$

$$\text{s.t.} \sum_{k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}})} \sum_{r \in \mathcal{R}_k} \delta_{irk} x_{rk} = 1, \qquad i \in \mathcal{N}_0, \tag{10b}$$

$$\sum_{r \in \mathcal{R}_k} x_{rk} \le M, \qquad k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}}), \tag{10c}$$

$$x_{rk} \ge 0, \qquad r \in \mathcal{R}_k, \quad k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}}). \tag{10d}$$

If M is chosen large enough (e.g., $M := N$), then the constraints (10c) are nonrestrictive and can be removed. We assume from now on that M is *nonrestrictive*; then, the model (10) is equivalent to the model (1a)–(1b), (4), except that the set of allowed vehicle types is smaller in the former. Hence, the column generation algorithm for the model (1a)–(1b), (4), described in Sect. 4.1, can be used to solve (10a)–(10b), (10d) [a solution to [BendersSP($\tilde{\mathbf{y}}$)] is then constructed by setting $x_{rk} := 0$ for $r \in \mathcal{R}_k$ and $k \in \mathcal{K} \setminus \tilde{\mathcal{K}}(\tilde{\mathbf{y}})$].

Now, let $L \ge 2$ denote the current Benders iteration and $\{1, \ldots, L-1\}$ the set of former iterations, and let $\tilde{\mathbf{y}}^L \in S$ be the fixed values of the variables $\mathbf{y}$ at iteration L ($\tilde{\mathbf{y}}^1 \in S$ are set manually). The model [BendersSP($\tilde{\mathbf{y}}^L$)] is solved by column generation applied to (10a)–(10b), (10d), and an optimal extreme point $(\boldsymbol{\pi}^L, \boldsymbol{\gamma}^L)$ to [BendersSPDual($\tilde{\mathbf{y}}^L$)] is calculated (details in Sect. 4.2.1), defining the constraint

$$v \ge \sum_{i \in \mathcal{N}_0} \pi_i^L + M \sum_{k \in \mathcal{K}} \gamma_k^L y_k, \tag{11}$$

to be added to the Benders restricted master problem, which is given by

$$\begin{aligned} [\text{BendersRMP}] \qquad \tilde{v}^L := \min_{v, \mathbf{y}} \quad & v, \\ \text{s.t.} \quad & v \ge \sum_{i \in \mathcal{N}_0} \pi_i^\ell + M \sum_{k \in \mathcal{K}} \gamma_k^\ell y_k, \quad \ell \in \{1, \ldots, L-1\}, \\ & \mathbf{y} \in S. \end{aligned}$$

After adding the constraint (11) to [BendersRMP], it is solved for optimal values $(\tilde{v}^{L+1}, \tilde{\mathbf{y}}^{L+1})$. This defines the new problem [BendersSP($\tilde{\mathbf{y}}^{L+1}$)]. This process is iterated—adding one constraint (11) in each Benders iteration—until an optimal solution to (2a)–(2d), (2f), (4) is found and verified.

With $(\tilde{v}^L, \tilde{\mathbf{y}}^L)$ optimal in [BendersRMP], $\tilde{\mathbf{x}}^L$ optimal in [BendersSP($\tilde{\mathbf{y}}^L$)], and $(\boldsymbol{\pi}^L, \boldsymbol{\gamma}^L)$ optimal in [BendersSPDual($\tilde{\mathbf{y}}^L$)], it follows that $(\tilde{\mathbf{x}}^L, \tilde{\mathbf{y}}^L)$ is an optimal solution to (2a)–(2d), (2f), (4) if it holds that

$$\tilde{v}^L = \sum_{i \in \mathcal{N}_0} \pi_i^L + M \sum_{k \in \mathcal{K}} \gamma_k^L \tilde{y}_k^L. \tag{12}$$

This is due to the fact that the inequalities $\tilde{v}^L \leq v^*$ [v^* being the optimal value of (2a)–(2d), (2f), (4)] and $\sum_{i\in\mathcal{N}_0} \pi_i^L + M\sum_{k\in\mathcal{K}} \gamma_k^L \tilde{y}_k^L = \sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} c_{rk}\tilde{x}_{rk}^L = w^*(\tilde{\mathbf{y}}^L) \geq v^*$ hold [where the second equality holds since $(\tilde{\mathbf{x}}^L, \tilde{\mathbf{y}}^L)$ is feasible in (2a)–(2d), (2f), (4)]. If Eq. (12) does not hold, then the inequality

$$\tilde{v}^L < \sum_{i\in\mathcal{N}_0} \pi_i^L + M\sum_{k\in\mathcal{K}} \gamma_k^L \tilde{y}_k^L$$

must hold. Therefore, by adding the constraint (11) to [BendersRMP], the current solution $(\tilde{v}^L, \tilde{\mathbf{y}}^L)$ becomes infeasible. Since the set S is finite, Benders' algorithm will converge after a finite[1] number of iterations.

Since the inequalities $\tilde{v}^L \leq w^L \leq w^*(\tilde{\mathbf{y}}^L)$, and possibly also $\tilde{v}^L = w^L < w^*(\tilde{\mathbf{y}}^L)$, hold, we have also implemented a stronger optimality criterion than (12), given by

$$\tilde{v}^L = \min_{\ell\in\{1,\ldots,L\}} \left\{ w^*(\tilde{\mathbf{y}}^\ell) \right\} =: w^L.$$

Benders' algorithm is then set to terminate when, for some $\varepsilon > 0$, the inequality $w^L - \tilde{v}^L \leq \varepsilon$ has become fulfilled, in which case the difference between the objective value of the best feasible solution to (2) found so far and that of an optimal solution to (2a)–(2d), (2f), (4) (i.e., $w^L - v^*$) is not greater than ε.

Since Benders' algorithm is applied to (2a)–(2d), (2f), (4), the optimal value of [BendersRMP] is a lower bound on that of (2), i.e., $\tilde{v}^L \leq z^*_{\text{EXT}}$. However, after solving [BendersSP($\tilde{\mathbf{y}}^L$)] binary requirements have been incurred on the variables x_{rk}, $r \in \tilde{\mathcal{R}}_k$, $k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}}^L)$, in [CGRMP] (when the column generation for (10a)–(10b), (10d) has converged). For the solution, $\tilde{\mathbf{x}}^L_{\text{binary}}$ say, to the resulting problem, it holds that $(\tilde{\mathbf{y}}^L, \tilde{\mathbf{x}}^L_{\text{binary}})$ is feasible in (2). Hence, the corresponding objective value is an upper bound on z^*_{EXT}.

4.2.1 An Optimal Extreme Point to the Benders Subproblem

In each Benders iteration the compact model (10a)–(10b), (10d) is solved in place of the equivalent model [BendersSP($\tilde{\mathbf{y}}$)]. Therefore, some extra effort must be put into finding an extreme point to the set $F_{\text{BendersSPDual}}$ that is also optimal in [BendersSPDual($\tilde{\mathbf{y}}$)].

[1] Finite convergence is guaranteed if an optimal solution $(\boldsymbol{\pi}^L, \boldsymbol{\gamma}^L)$ to [BendersSPDual($\tilde{\mathbf{y}}^L$)] is used to generate a new constraint to [BendersRMP] in each iteration, since if $\tilde{\mathbf{y}}^{\ell_1} = \tilde{\mathbf{y}}^{\ell_2}$, for two Benders iterations $\ell_1 < \ell_2$, then the optimality criterion (12) will be fulfilled at iteration ℓ_2.

The final column generation iteration in the solution course for the model (10a)–(10b), (10d) yields optimal variable values, $\bar{\boldsymbol{\pi}}$, for its dual, which is expressed as

$$\max_{\boldsymbol{\pi}} \quad \sum_{i \in \mathcal{N}_0} \pi_i, \tag{13a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{N}_0} \delta_{irk} \pi_i \leq c_{rk}, \qquad r \in \mathcal{R}_k, \quad k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}}). \tag{13b}$$

For $k \in \mathcal{K}$ and $\bar{\boldsymbol{\pi}} \in \mathbb{R}^N$ we define the problem

$$\gamma_k^* := \max_{\gamma_k} \left\{ \gamma_k \leq 0 \;\middle|\; \gamma_k \leq c_{rk} - \sum_{i \in \mathcal{N}_0} \delta_{irk} \bar{\pi}_i,\ r \in \mathcal{R}_k \right\} \tag{14a}$$

$$= \min \left\{ 0; \min_{r \in \mathcal{R}_k} \left\{ c_{rk} - \sum_{i \in \mathcal{N}_0} \delta_{irk} \bar{\pi}_i \right\} \right\}, \tag{14b}$$

which is closely related to [BendersSPDual($\tilde{\mathbf{y}}$)]: The optimal values $\boldsymbol{\gamma}^*$ equal the maximum possible values of $\boldsymbol{\gamma}$ in [BendersSPDual($\tilde{\mathbf{y}}$)] for $\boldsymbol{\pi} := \bar{\boldsymbol{\pi}}$. We have the following result, the proof of which is found in [12, Sect. 4.2.2].

Proposition 1. *Consider the following properties of a vector* $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}}) \in \mathbb{R}^{N+K}$.

(i) $\bar{\boldsymbol{\pi}}$ *is optimal in* (13).
(ii) $\bar{\boldsymbol{\pi}}$ *is an extreme point to the set* $\left\{ \boldsymbol{\pi} \in \mathbb{R}^N \;\middle|\; \boldsymbol{\pi} \text{ satisfies (13b)} \right\}$.
(iii) $\bar{\gamma}_k = 0$ *for all* $k \in \tilde{\mathcal{K}}(\tilde{\mathbf{y}})$.
(iv) $\bar{\gamma}_k = \gamma_k^*$, *where* γ_k^* *is optimal in* (14), *for all* $k \in \mathcal{K} \setminus \tilde{\mathcal{K}}(\tilde{\mathbf{y}})$.

The following statements hold for a vector $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}}) \in \mathbb{R}^{N+K}$.

(a) If (i), (iii), and (iv) hold, then $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}})$ *is optimal in [BendersSPDual($\tilde{\mathbf{y}}$)].*
(b) If (ii), (iii), and (iv) hold, then $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}})$ *is an extreme point to the set* $F_{\text{BendersSPDual}}$.

After solving (10a)–(10b), (10d)—the solution of which, with the excluded variables set to zero, is optimal in [BendersSP($\tilde{\mathbf{y}}$)]—using column generation, values of $\bar{\boldsymbol{\pi}}$ satisfying property (i) results from the last iteration.

The important consequences of Proposition 1 are the following: If $\bar{\boldsymbol{\gamma}}$ fulfills properties (iii) and (iv) then $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}})$ is optimal in [BendersSPDual($\tilde{\mathbf{y}}$)]. If $\bar{\boldsymbol{\pi}}$ also fulfills property (ii)—implying that $\bar{\boldsymbol{\pi}}$ is an optimal extreme point to (13)—and $\bar{\boldsymbol{\gamma}}$ fulfills properties (iii) and (iv) for the chosen values of $\bar{\boldsymbol{\pi}}$, then $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}})$ is an extreme point to $F_{\text{BendersSPDual}}$ that is optimal in [BendersSPDual($\tilde{\mathbf{y}}$)]. Note that the values $\bar{\boldsymbol{\gamma}}$ that fulfill properties (iii) and (iv) are given by $\bar{\gamma}_k = \min\left\{0; \min_{r \in \mathcal{R}_k} \{\hat{c}_{rk}\}\right\}$, which equals the optimal value γ_k^* of (14) for all $k \in \mathcal{K}$.

In the context of the LP model (1a)–(1b), (4), $\hat{c}_{rk}$ is the reduced cost of the variable x_{rk} [cf. the model (14) and the reduced cost (6)]. Hence, according to Proposition 1, the right-hand side of each constraint (11) that is added to

[BendersRMP] equals the sum of $\sum_{i\in\mathcal{N}_0}\pi_i^L$ (being the optimal objective value of the Benders subproblem in iteration L) and $M\sum_{k\in\mathcal{K}}\gamma_k^L y_k$ [the sum of the scaled minimal reduced costs, γ_k^L, from the Benders iteration L, for all vehicle types in the set $\tilde{\mathcal{K}}(\mathbf{y})$]. So when solving [BendersRMP] in iteration L, the resulting variable values $\tilde{\mathbf{y}}^L$ are the most promising with respect to the objective value $\sum_{i\in\mathcal{N}_0}\pi_i^\ell$ plus the corresponding minimal reduced costs $\bar{\gamma}_k^\ell$ (scaled by M), $k\in\tilde{\mathcal{K}}(\tilde{\mathbf{y}}^L)$, taking into account each former Benders iteration $\ell\in\{1,\ldots,L-1\}$.

4.2.2 Extensions of Benders' Algorithm

To improve the speed of convergence of Benders' algorithm, we have implemented the following *projection-of-routes* procedure, in which optimal solutions, $\tilde{\mathbf{x}}^\ell$, to [BendersSP($\tilde{\mathbf{y}}^\ell$)] are investigated to determine whether the corresponding routes can yield a better objective value when used by other types of vehicles. Since only vehicle types in the set $\tilde{\mathcal{K}}(\tilde{\mathbf{y}}^\ell)$ are allowed in [BendersSP($\tilde{\mathbf{y}}^\ell$)], it may occur that a route that is used in an optimal solution to [BendersSP($\tilde{\mathbf{y}}^\ell$)] can be taken by a vehicle type in $\mathcal{K}\setminus\tilde{\mathcal{K}}(\tilde{\mathbf{y}}^\ell)$ at a lower cost than that corresponding to any of the vehicle types in $\tilde{\mathcal{K}}(\tilde{\mathbf{y}}^\ell)$. Consider a subset $\hat{\mathcal{K}}\subset\mathcal{K}$ such that $\left|\hat{\mathcal{K}}\right|\leq C$. If, for some $\ell=1,\ldots,L$, the routes in an optimal solution to [BendersSP($\tilde{\mathbf{y}}^\ell$)] can be assigned to the vehicle types in $\hat{\mathcal{K}}$ at a cost that is lower than w^L (i.e., the lowest value of a feasible solution found so far), then $\hat{\mathcal{K}}$ is used to define the variable values $\tilde{\mathbf{y}}^{L+1}$ in the next iteration. Thus, instead of solving [BendersRMP] for $(\tilde{v}^{L+1},\tilde{\mathbf{y}}^{L+1})$, we set

$$\tilde{y}_k^{L+1}:=\begin{cases}1, & \text{if } k\in\hat{\mathcal{K}},\\ 0, & \text{otherwise.}\end{cases}$$

Our implementation of the selection of the set $\hat{\mathcal{K}}$ is described in [12, Sect. 4.2.4].

The projection-of-routes procedure, which proved to greatly reduce the time until an optimal solution to (2a)–(2d), (2f), (4) is found, is set to be performed each Benders iteration, starting from iteration three, and solutions to former [BendersSP($\tilde{\mathbf{y}}$)] that are tested is restricted to the 100 latest Benders iterations.

To reduce the solution time of Benders subproblems, a so-called *warm start* can be utilized in which the routes that are part of optimal solutions to former Benders subproblems, and which can also be taken by some vehicle in the current Benders iteration, are included in the set of routes used to initialize the column generation algorithm for solving the Benders subproblem. This has greatly reduced the time spent in the column generation phase. In many Benders iterations, even an optimal solution to the Benders subproblem is found among the routes provided in the warm start.

4.2.3 Suggestions for Further Improvements of Benders' Algorithm

To obtain an optimal extreme point to $F_{\text{BendersSPDual}}$, according to Proposition 1 the Benders subproblem should first be solved to optimality using column generation, yielding an optimal extreme point $\bar{\boldsymbol{\pi}}$ to (13). While in the later column generation iterations typically only very small improvements are made, a constraint (11) of high quality can be provided by a point that is not necessarily *extreme* in the set $F_{\text{BendersSPDual}}$ and *near-optimal* in the current Benders subproblem. For any $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}}) \in F_{\text{BendersSPDual}}$ a valid inequality for [BendersRMP] is given by (see [5, p. 308])

$$v \geq \sum_{i \in \mathcal{N}_0} \bar{\pi}_i + M \sum_{k \in \mathcal{K}} \bar{\gamma}_k y_k.$$

Thus, the column generation algorithm for the Benders subproblem can be terminated before an optimal extreme point to (13) is found, and the variable values $\bar{\boldsymbol{\pi}}$ that are optimal in the current [CGRMPDual] can be used to define a new constraint (11) due to the fact that any point $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\gamma}}) \in \mathbb{R}^{N+K}$ such that $\bar{\boldsymbol{\gamma}} \leq \boldsymbol{\gamma}^*$ (where γ_k^* is optimal in (14), $k \in \mathcal{K}$) belongs to the set $F_{\text{BendersSPDual}}$.

Instead of solving the problems (14), $k \in \mathcal{K}$, in each Benders iteration [as prescribed in Proposition 1(a)] lower bounds on $\boldsymbol{\gamma}^*$ can be used to define a new constraint (11). This may greatly reduce the computational effort required, since the problems (14) are essentially equivalent to the computationally expensive column generation subproblems (i.e., the ESPPRC; see Sect. 4.1.1) for the model (1). Dynamic programming applied to the SPPRC with 2-cycle elimination is an efficient and commonly used method for solving relaxed column generation subproblems in connection with the VRP [13, p. 417]; it also yields good-quality lower bounds on the values $\boldsymbol{\gamma}^*$. This usage of lower bounds was employed in [6] in a column generation solution approach for the HVRP; it has not been implemented here, but is suggested as a means for improving our algorithm.

5 Tests and Results

The tests of our algorithms were performed on a Linux computer with a Pentium Dual-Core CPU 2.5 GHz with 2,048 KB cache. The mathematical models and algorithms were implemented using the modeling software AMPL and the optimization solver CPLEX 12. Dynamic programming for the column generation subproblems and some other calculations (e.g., finding the initial solutions to [CGRMP], the warm start, and the projection-of-routes) were implemented in Matlab.

5.1 Test Settings

The instances CT12 were constructed as follows. First, the instances G12 were constructed using the instances in [7]—these were downloaded from [10]—following the instructions in [16]. The networks of customer nodes of the instances CT12 are equivalent to those of G12. The capacities and the fixed and variable costs for the vehicle types of the instances CT12 were taken from [6, Table 1] (each instance contains three to six vehicle types). A smaller vehicle is always less expensive than a larger one, i.e., for any two indices $k_1, k_2 \in \mathcal{K}$ such that $D_{k_1} < D_{k_2}$ holds, the inequalities $f_{k_1} < f_{k_2}$ and $\alpha_{k_1} < \alpha_{k_2}$ hold.

To test our models for the many-FSMVRP, we then extended the instances 3–6 in CT12 (with $N = 20$) and the instance 13 in G12 (with $N = 50$) to include a larger set of vehicle capacities within the ranges of their respective original capacities. Since all customer demands are integer valued, we included all the integer values in each capacity interval (see the Appendix for details). We denote these individual test instances by mFSM-3,…,mFSM-6, and mFSM-13, following the numbering in [16]; the collection of instances is denoted by many-FSMVRP5. Each of the instances in many-FSMVRP5 comprises 91–181 vehicle types (e.g., for mFSM-3, $K = 101$ and $D_k \in \{20, \ldots, 120\}$), which are numbered such that $k_1 < k_2$ implies $D_{k_1} < D_{k_2}$.

Our tests are performed using the following algorithmic settings. The column generation algorithm is applied to the relaxation (1a)–(1b), (4) of the model (1); it is in this section abbreviated as *column generation*. Benders' algorithm, with column generation used to solve the Benders subproblems, is applied to the relaxation (2a)–(2d), (2f), (4) of the extended model (2); it is in this section abbreviated as *Benders' algorithm* and is applied *with* or *without the projection-of-routes* procedure.

5.2 Comparison of the Algorithms

We first compare the column generation, and Benders' algorithm with and without the projection-of-routes, using the four smallest instances, mFSM-3, …, mFSM-6. The comparison is enabled by the choice of a nonrestrictive vehicle type limit C in the extended model, such that the two relaxed models share an optimal solution and $z^*_{\text{LP}} = v^*$. Benders' algorithm has been initiated with those vehicle types that were included in the corresponding instances in CT12. Detailed results for the column generation, and for Benders' algorithm with the projection-of-routes, are presented in Tables 1 and 2, respectively.

Benders' algorithm without the projection-of-routes procedure performs better than the column generation algorithm for mFSM-4 and mFSM-6, but quite a lot worse for mFSM-3 and mFSM-5.

Table 1 Results obtained by the column generation

Instance	$\bar{z}$	z^*_{LP}	T (CPU s)	$T_{z^*_{\mathrm{LP}}}$ (CPU s)	$(\bar{z}-z^*_{\mathrm{LP}})/z^*_{\mathrm{LP}}$ (%)
mFSM-3	1,010.0	1,010.0	2,376	2,310	0
mFSM-4	6,550.7	6,366.0	4,608	4,607	2.90
mFSM-5	1,187.0	1,180.6	3,066	2,285	0.54
mFSM-6	6,691.7	6,436.6	3,681	3,650	3.96

$\bar{z}$ denotes the value of the best feasible solution to (1) found (at the last column generation iteration, with the binary requirements (1c) reinserted), and z^*_{LP} denotes the optimal value of (1a)–(1b), (4). Note that the inequalities $z^*_{\mathrm{LP}} \leq z^* \leq \bar{z}$ and $z^*_{\mathrm{LP}} \leq \tilde{z}_{\mathrm{CGRMP}} \leq \bar{z}$ hold, where z^* denotes the optimal value of (1). The total computation time in AMPL/CPLEX and Matlab is denoted by T, and $T_{z^*_{\mathrm{LP}}}$ denotes the computation time of the column generation algorithm until an optimal solution to (1a)–(1b), (4) is found

Table 2 Results obtained by Benders' algorithm applied with the projection-of-routes

Instance	C [a]	$\bar{z}_{\mathrm{EXT}}$	w^L	$\tilde{v}^L$	T (CPU s)	T_{v^*} (CPU s)	L	$(\bar{z}_{\mathrm{EXT}}-v^*)/v^*$ (%)
mFSM-3	10	1,010.0	1,010.0	900.7	9,963	2,078	100	0
mFSM-4	4	6,484.7	6,366.0	6,366.0	1,107	980	3	1.86
mFSM-5	13	1,188.7	1,180.6	1,089.6	58,801	9,666	100	0.69
mFSM-6	6	6,580.8	6,436.6	6,436.6	1,300	1,140	4	2.24

C denotes the (nonrestrictive) vehicle type limit. $\bar{z}_{\mathrm{EXT}}$ denotes the value of the best solution found, when reinserting the binary restrictions in the last iteration of the column generation applied to [BendersSP($\tilde{\mathbf{y}}^L$)]. w^L ($\geq v^*$) is the best value of [BendersSP($\tilde{\mathbf{y}}^\ell$)] obtained (actually, $w^L = v^*$ for each of the four instances). $\tilde{v}^L$ is the optimal value of [BendersRMP] in the Benders iteration L, and v^* is the optimal value of (2a)–(2d), (2f), (4). Note that the inequalities $\tilde{v}^L \leq v^* \leq z^*_{\mathrm{EXT}} \leq \bar{z}_{\mathrm{EXT}}$ hold, where z^*_{EXT} denotes the optimal value of (2). The total computation time in AMPL/CPLEX and Matlab is denoted by T, and T_{v^*} denotes the computation time for the Benders iterations until the optimal solution to (2a)–(2d), (2f), (4) is found. L is the number of Benders iterations performed (maximally 100) until the optimality is verified. Since the vehicle type limit C is nonrestrictive, $v^* = z^*_{\mathrm{LP}}$ holds (see Table 1)

[a] $C := \sum_{k\in\mathcal{K}} \left\lceil N^{-1} \sum_{r\in\mathcal{R}_k} \bar{x}^*_{rk} \right\rceil + 2$, where $\bar{\mathbf{x}}^*$ denotes an optimal solution to (1a)–(1b), (4)

For the instances mFSM-4 and mFSM-6, Benders' algorithm without the projection-of-routes converges to an optimal solution to (2a)–(2d), (2f), (4) [which is optimal also to (1a)–(1b), (4)] in just two and three iterations, respectively, taking less computing time than the column generation approach for the model (1).

For none of the instances mFSM-3 and mFSM-5, Benders' algorithm without the projection-of-routes manages to pick a combination of vehicle types that is used in an optimal solution to (2a)–(2d), (2f), (4), not even after 100 Benders iterations, which calls for a lot more computing time than does the column generation approach for the model (1). Each of these vehicle types is, however, chosen quite frequently, which is illustrated in Fig. 1 for the instance mFSM-3. For mFSM-4 and mFSM-6, the cost structure is such that only vehicle types among the five smallest are used in the respective optimal solutions to (2a)–(2d), (2f), (4), while there is a larger spread of the optimal vehicle types for the instances mFSM-3 and mFSM-

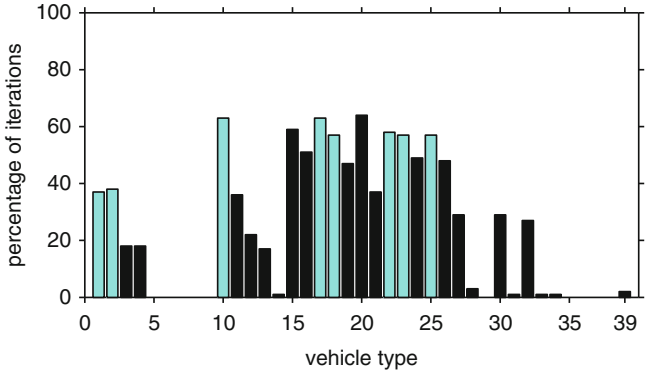


Fig. 1 The percentage of 100 Benders iterations without the projection-of-routes, in which each of the vehicle types (1–101) is picked, for mFSM-3 with $C = 10$. The optimal solution uses vehicles in $\tilde{\mathcal{K}}(\tilde{\mathbf{y}}^L) = \{1, 2, 10, 17, 18, 22, 23, 25\}$ (*light bars*). Although $\tilde{\mathcal{K}}(\tilde{\mathbf{y}}^1) = \{1, 11, 21, 51, 101\}$, only vehicle types among the 39 smallest are picked in any Benders iteration

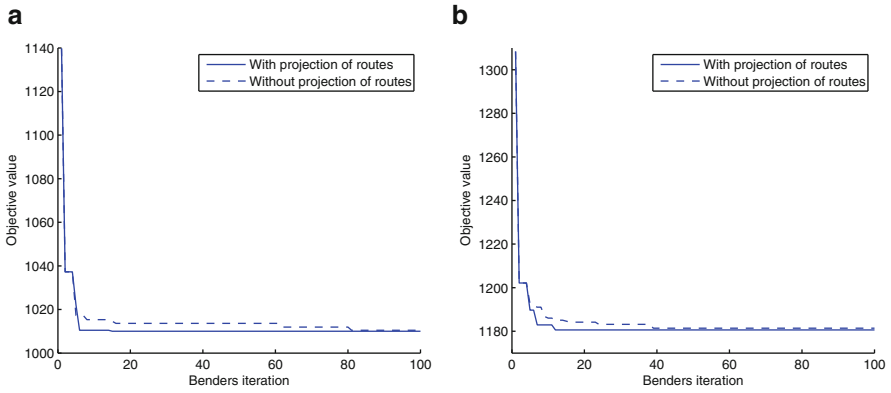


Fig. 2 The best objective value, w^L, of (2a)–(2d), (2f), (4) found at Benders iteration L. (**a**) The instance mFSM-3; when using the projection-of-routes, $w^{15} = 1{,}010.0 = v^*$; without the projection-of-routes, $w^{100} = 1{,}010.5$. (**b**) The instance mFSM-5; when using the projection-of-routes, $w^{12} = 1{,}180.6 = v^*$; without the projection-of-routes, $w^{100} = 1{,}181.4$

5; this may explain the difference in convergence speed. Benders' algorithm with the projection-of-routes, however, finds optimal solutions to the instances mFSM-3 and mFSM-5 within 15 Benders iterations.

When the projection-of-routes is not employed, for the instance mFSM-3, the best objective value, $w^L = 1{,}010.5$, of the Benders subproblem over 100 Benders iterations is attained at iteration 81. When employing the projection-of-routes, this value is attained at iteration six and the optimal value of (2a)–(2d), (2f), (4), $v^* = 1{,}010.0$, is attained at iteration 15. Similar results are obtained for mFSM-5 (see Fig. 2).

Benders' algorithm, with the projection-of-routes, finds the optimal value, v^*, of (2a)–(2d), (2f), (4), for the instances mFSM-3,...,mFSM-6. For the instances mFSM-3, mFSM-4, and mFSM-6, it attains the optimal value faster than does the column generation algorithm. Verifying optimality for the instances mFSM-3 and mFSM-5 using Benders' algorithm, however, calls for a very long computing time (our computations were interrupted after more than 1 week).

We also compared our column generation algorithm applied to (1) with the column generation of Choi and Tcha in [6], using the instances in CT12. Employing the relaxed column generation subproblems (i.e., SPPRC), a set-covering model in place of the set-partitioning model (1), and an implementation in C, the latter algorithm definitely outperforms our column generation implementation (it finds better solutions and is at least two orders of magnitude faster than our method). This suggests that great improvements can be made to our implementations of the column generation as well as of Benders' algorithm, since the latter uses column generation for solving the subproblems. Another interesting result is that the solution times seem to scale quite well with an increasing size of the sets of vehicle types (while increasing the sizes of the sets of vehicle types with an average factor of 25, the solution times increased with an average factor of six).

5.3 Comparison of the Solutions Obtained Using Different Models

When the aim is to choose a limited number of vehicle types from a very large set, the model (1) cannot be used. The possibility to impose such a limit constitutes a valuable property of the extended model (2) and may result in a more flexible fleet.

Figure 3 illustrates the best feasible solutions found for the instance mFSM-13, (a) with an unlimited number of vehicle types, obtained by the column generation, and (b) with the vehicle type limit $C = 4$, obtained by Benders' algorithm without the projection-of-routes. For the limited case, $\bar{z}_{\mathrm{EXT}} = 2{,}834.4$ and four vehicle types with $D_k \in [18, 31]$ are used, whereas for the unlimited case, $\bar{z} = 2{,}753.6$ and 16 vehicle types with $D_k \in [1, 32]$ are used. Hence, for mFSM-13, out of the totally 181 vehicle types, only vehicles among the 32 smallest types are used in each case. The corresponding instance in CT12, which includes totally only six vehicle types, possesses the optimal objective value 2,964.7; see [1, Table 7].

We have compared the column generation with Benders' algorithm, without the projection-of-routes, employing load-dependent costs according to (3) for the instance mFSM-13. The parameter values $Q^{\mathrm{dist}} := 1.4$ and $Q^{\mathrm{load}} := 0.05$ chosen yielded the most reasonable solutions among the values tried when applying the column generation. The corresponding best solutions found are illustrated in Fig. 4. For ease of comparison, the load-dependent objective values have been converted to the original costs, by calculating the cost of the optimal route–vehicle pairs (r, k) using (1a) instead of (3). Apparently, smaller vehicle types are used to a larger extent

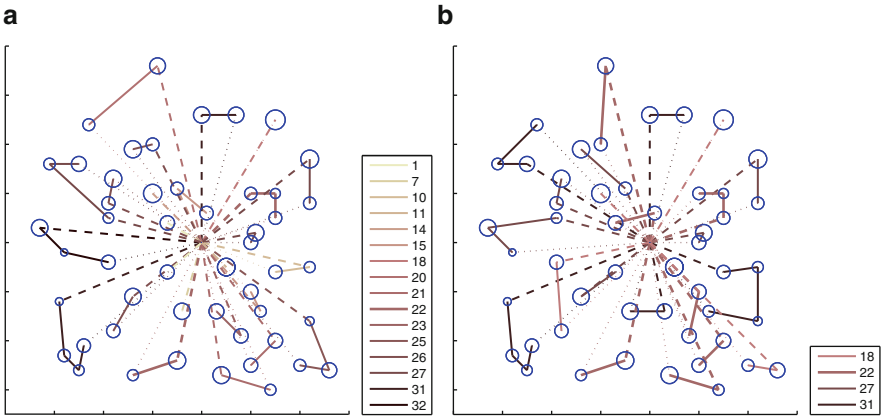


Fig. 3 Illustrations of the best solutions found for the instance mFSM-13, employing the original costs c_{rk}, without/with a vehicle type limit. The depot is centrally located among the customer nodes, whose areas are proportional to their respective demand. The routes are marked by a *dashed line* for the first link (leaving the depot), *solid lines* for intermediate links, and a (*thin*) *dotted line* for the last link (returning to the depot). Different *colors* represent different vehicle types. Here, $z^*_{\mathrm{LP}} = 2{,}748.9 \leq z^* \leq z^*_{\mathrm{EXT}}$. (**a**) The best solution found using column generation. Here, $\bar{z} = 2{,}753.6 \geq z^*$. (**b**) The best solution found using Benders' algorithm. Here, $C = 4$ and $\bar{z}_{\mathrm{EXT}} = 2{,}834.4 \geq z^*_{\mathrm{EXT}}$

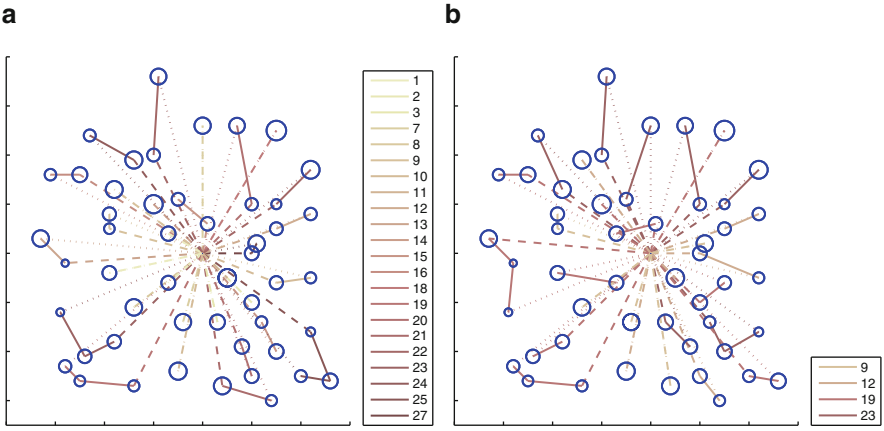


Fig. 4 Illustrations of the best solutions found for the instance mFSM-13, using the load-dependent costs c^{load}_{rk}, without/with a vehicle type limit. For an interpretation of the plot, see Fig. 3. Only vehicles among the 27 smallest (out of 181) types are used in each solution. Here, $z^*_{\mathrm{LP}} = 2{,}801.9$. (**a**) The best solution found using column generation. Here, $\bar{z} = 2{,}808.1 \geq z^*$. (**b**) The best solution found using Benders' algorithm. Here, $C = 4$ and $\bar{z}_{\mathrm{EXT}} = 2{,}883.8 \geq z^*_{\mathrm{EXT}}$

for the case of load-dependent costs (Fig. 4) compared to that of load-independent costs (Fig. 3). Since the vehicle is empty when returning to the depot, routes, for which the last link is long compared with the total length of the route, are more common for the case of load-dependent costs.

Interestingly, the total solution times are generally shorter for the case of load-dependent costs than for that of load-independent costs; this may depend on the lack of symmetries in the former case. An instance of the VRP with *time-windows*, possessing *tight* time-windows, may be solved faster than the corresponding instance without time-windows; see [4]. Similarly, the load-dependent costs may lead to a more constrained problem, which may also explain the shorter solution times.

6 Conclusions

We have extended the *fleet size and mix vehicle routing problem* (FSMVRP) to include a *large set of vehicle types*, then denoted many-FSMVRP. Further, we have developed and tested mathematical models and algorithms for the many-FSMVRP. The results of the test performed on a set of instances indicate that the extended set-partitioning model solved using Benders' algorithm performs on par with the set-partitioning model solved using column generation (when the two models share an optimal solution), with the exception that for some of the instances Benders' algorithm requires a very long time to verify the optimality. The extended set-partitioning model with a restrictive limit on the number of vehicle types—for which Benders' algorithm is required—yields solutions with different characteristics than with a nonrestrictive limit. When extending the many-FSMVRP to include load-dependent costs—hence increasing the practical applicability of the model—solutions with different characteristics were found; the solution times were, however, not greatly impacted.

The advantages of using Benders' algorithm for the extended set-partitioning model are indicated by the following: a clear pattern emerges in which some vehicle types, which are part of an optimal set of vehicle types, are chosen more often than other types. Each constraint that is added to the Benders restricted master problem has a nice interpretation as the objective value of the optimal solution to the Benders subproblem plus a weighted sum of reduced costs (see Sect. 4.2.1); hence, the collected information gained from the reduced costs of solutions to former Benders subproblems is used when new vehicle types are chosen in each Benders iteration; this provides a guide for choosing new vehicle types.

Also, the set of optimal routes for a given problem instance depends to a large extent on the available vehicle types. The improvements gained by using the projection-of-routes procedure suggest that Benders' algorithm succeeds in choosing vehicle-type configurations which yield Benders subproblems whose optimal solutions are composed by high-quality routes. Using only those routes that are part of former optimal solutions to Benders subproblems, the projection-of-routes is found to consistently determine an optimal combination of vehicles and routes. Thus, for problem instances with a considerably larger set of vehicle types than the instances in many-FSMVRP5, we suggest the following approach. Temporarily restrict the set of vehicle types and—similarly to the projection-of-routes, in which

vehicle types that are not part of the restricted set are allowed to be matched with routes from solutions to the Benders subproblems—update the set using the information from the former Benders iterations, including vehicle types that are similar to those that have been chosen often and excluding vehicle types that have not been chosen.

To improve the performance of the proposed algorithms we propose the following adjustments. For the column generation algorithms (which are employed also within Bender's algorithm), the dynamic programming solution of the subproblems should gain from an improved implementation. Also, the column generation subproblems should be relaxed, as suggested in Sect. 4.1.1, and the set-partitioning constraints should be relaxed to set-covering constraints. The results in [6] indicate that these changes would result in shorter computing times and solutions with lower objective values. For Benders' algorithm, the relaxation suggested for the column generation subproblems should be applied to the problems (14), which are solved for each vehicle type k in each Benders iteration (see Sect. 4.2.3).

The models should also be altered to allow for the inclusion of more properties of real problem settings; see [17] for numerous possible extensions. In [4] a HVRP with multiple depots and time-windows is solved using a branch-and-cut-and-price heuristic, with a dynamic programming solution of the column generation subproblems. The authors state that it is probably the heterogeneous fleet that "really complicates the problem" (see [4, p. 735]), but also that their implementation often performs better than other heuristics from the literature. Heuristics based on mathematical programming techniques are becoming increasingly popular; see [11]. The findings in [4], along with the strong competitiveness for the HVRP of the column generation algorithm in [6], indicate that the models and algorithms developed and presented in this chapter may be competitive as heuristics for the many-FSMVRP.

Appendix: The Extended Test Instances many-FSMVRP5

The sets of vehicle types of the instances in CT12 are extended as follows: the fixed and variable costs are defined using the spline function in Matlab according to

```
fcostEXT = interp1(capacity, fcost, capacityEXT,
     'spline');
vcostEXT = interp1(capacity, vcost, capacityEXT,
     'spline');
```

Here, `capacity` denotes a vector with the capacities of the original vehicle types, `fcost` (`vcost`) denotes a vector with the fixed (variable) costs of the original vehicle types, `capacityEXT` denotes a vector with the capacities of the extended set of vehicle types, and `fcostEXT` (`vcostEXT`) denotes the resulting vector with the fixed (variable) costs of the corresponding extended set of vehicle types.

For each of the instances in many-FSMVRP5, a vehicle type is removed if no route in the set $\cup_{k\in\mathcal{K}}\mathcal{R}_k$ has a total customer demand that is equal to the capacity of this vehicle type (except for the smallest capacity). This removal of a vehicle type does not exclude any optimal solution from the feasible set, since any route assigned to such a vehicle can be assigned to a smaller vehicle at a lower cost. No vehicle type was, however, removed from any of the instances in CT12.

References

1. Baldacci, R., Mingozzi, A.: A unified exact method for solving different classes of vehicle routing problems. Math. Programm. **120**(2), 347–380 (2009)
2. Baldacci, R., Battarra, M., Vigo, D.: Routing a heterogeneous fleet of vehicles. In: Golden, B.L., Raghavan, S., Wasil, E.A. (eds.) The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces, vol. 43, pp. 3–27. Springer, New York (2008)
3. Balinski, L.M., Quandt, R.E.: On an integer program for a delivery problem. Oper. Res. **12**(2), 300–304 (1964)
4. Bettinelli, A., Ceselli, A., Righini, G.: A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. Transp. Res. C **19**(5), 723–740 (2011)
5. Boschetti, M., Maniezzo, V.: Benders decomposition, Lagrangean relaxation and metaheuristic design. J. Heuristics **15**(3), 283–312 (2009)
6. Choi, E., Tcha, D-W.: A column generation approach to the heterogeneous fleet vehicle routing problem. Oper. Res. **34**(7), 2080–2095 (2007)
7. Christofides, N., Eilon, S.: An algorithm for the vehicle-dispatching problem. OR **20**(3), 309–318 (1969)
8. Cordeau, J., Stojković, G., Soumis, F., Desrosiers, J.: Benders decomposition for simultaneous aircraft routing and crew scheduling. Transp. Sci. **35**(4), 375–388 (2001)
9. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manag. Sci. **6**(1), 80–91 (1959)
10. Díaz, B.D.: The VRP Web: VRP Instances, Nov 2006
11. Drexl, M.: Rich vehicle routing in theory and practice. Logist. Res. **5**(1–2), 47–63 (2012)
12. Eriksson Barman, S.: Modeling and solving vehicle routing problems with many available vehicle types. Master's thesis, University of Gothenburg, Sweden (2014)
13. Feillet, D.: A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR **8**(4), 407–424 (2010)
14. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C.: An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks **44**(3), 216–229 (2004)
15. Glover, F., Laguna, M.: Tabu search. In: Pardalos, P.M., Du, D., Graham, R.L. (eds.) Handbook of Combinatorial Optimization, pp. 3261–3362. Springer, SIAM Publishing, New York (2013)
16. Golden, B.L., Assad, A., Levy, L., Gheysens, F.: The fleet size and mix vehicle routing problem. Comput. Oper. Res. **11**(1), 49–66 (1984)
17. Hoff, A., Andersson, H., Christiansen, M., Hasle, G., Løkketangen, A.: Industrial aspects and literature survey: fleet composition and routing. Comput. Oper. Res. **37**(12), 2041–2061 (2010)
18. Irnich, S., Desaulniers, G.: Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) Column Generation, pp. 33–65. Springer, New York (2005)
19. Lasdon, L.S.: Optimization Theory for Large Systems. Macmillan, London (Reprinted by Dover Publications, Mineola, NY, 2002) (1970)

20. Lübbecke, M.E., Desrosiers, J.: Selected topics in column generation. Oper. Res. **53**(6), 1007–1023 (2005)
21. Pessoa, A., Poggi de Aragão, M., Uchoa, E.: A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. In: Demetrescu, C. (ed.) Experimental Algorithms. Lecture Notes in Computer Science, vol. 4525, pp. 150–160. Springer, Berlin (2007)
22. Righini, G., Salani, M.: New dynamic programming algorithms for the resource constrained shortest path problem. Networks **51**(3), 155–170 (2008)
23. Taillard, E.D.: A heuristic column generation method for the heterogeneous fleet VRP. RAIRO: Oper. Res. **33**(1), 1–14 (1999)
24. Toth, P., Vigo, D.: An overview of vehicle routing problems. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, pp.1–26. SIAM Publishing, Philadelphia, PA (2002)
25. Toth, P., Vigo, D.: Preface. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, pp. xvii–xviii. SIAM Publishing, Philadelphia, PA (2002)
26. Vanderbeck, F.: On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. Oper. Res. **48**(1), 111–128 (2000)
27. Vanderbeck, F.: Implementing mixed integer column generation. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) Column Generation, pp. 331–358. Springer, New York (2005)
28. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: Heuristics for multi-attribute vehicle routing problems: a survey and synthesis. Eur. J. Oper. Res. **231**(1), 1–21 (2013)
29. Xiao, Y., Zhao, Q., Kaku, I., Xu, Y.: Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Comput. Oper. Res. **39**(7), 1419–1431 (2012)

A Genetic Algorithm for Scheduling Alternative Tasks Subject to Technical Failure

Dalila B.M.M. Fontes and José Fernando Gonçalves

Abstract Nowadays, organizations are often faced with the development of complex and innovative projects. This type of projects often involves performing tasks which are subject to failure. Thus, in many such projects several possible alternative actions are considered and performed simultaneously. Each alternative is characterized by cost, duration, and probability of technical success. The cost of each alternative is paid at the beginning of the alternative and the project payoff is obtained whenever an alternative has been completed successfully. For this problem one wishes to find the optimal schedule, i.e., the starting time of each alternative, such that the expected net present value is maximized. This problem has been recently proposed in Ranjbar (Int Trans Oper Res 20(2):251–266, 2013), where a branch-and-bound approach is reported. Since the problem is NP-*Hard*, here we propose to solve the problem using genetic algorithms.

Keywords Scheduling under activity failure • Maximization of expected net present value • Biased random-key genetic algorithms

1 Introduction

Companies must plan and optimize their activities in a uncertain environment. The uncertainties may come from several different parts of their business. The uncertainties most commonly addressed in the literature are related to the costs and returns associated with the business. Regarding scheduling problems the most frequently studied uncertainties are resource breakdowns and duration variability. However, other sources of uncertainty exist. For example, Research and Development (R&D) companies, highly dependent on innovation, also face uncertainty regarding the success of their initiatives. These initiatives, usually called projects, may fail. Thus, in order to deal with this kind of uncertainty companies may have

D.B.M.M. Fontes (✉) • J.F. Gonçalves
Faculdade de Economia da Universidade do Porto and LIAAD INESC TEC,
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
e-mail: fontes@fep.up.pt; jfgoncal@fep.up.pt

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_7

to consider several alternative ways of developing their projects (see, e.g., [27, 28]). In this type of projects, the alternatives are of the same kind, although different, and pursue a similar goal. For example, their execution may represent the repetition of trials until success in one is achieved. Usually, the alternatives are related and some alternatives may imply the execution of some other alternatives, i.e., there are precedence relations between some of the alternatives.

This work addresses the scheduling of alternatives subject to technical failure, in order to maximize the expected Net Present Value (NPV) of the project. The NPV of a project is the discounted value of the project cash flows. The NPV is affected by the project schedule and in capital-intensive industries, the timing of expenditures has a major impact on project feasibility and profitability.

Most of the relevant sources of literature considering activity failure come from chemical engineering applications, where Grossmann and his colleagues have been addressing such problems. In [25] a mixed integer linear programming model was proposed to schedule the activities of a single product considering precedence constraints. Activities have associated a cost, a duration, and a probability of success. The objective was to minimize the expected cost. This model was subsequently used on a specific application [26]. In [19] the authors propose a two-stage stochastic optimization approach to account for the uncertainty in the outcome of the trials. A recent survey on optimization challenges and opportunities in the pharmaceutical industry can be found in [21].

Other scheduling problems involving activity failures have been addressed, see the survey in [8]. De Reyck and colleagues study the scheduling of activities with uncertain outcomes, where project success is achieved only if all individual activities succeed. In [7], the authors have considered the project scheduling problem with uncertain activity outcomes and known durations. This work was extended in [4, 5] where activity durations are stochastic. More recently, in [2] the scheduling of projects subject to failure has been considered. In this problem, several projects, each consisting of several activities, have to be scheduled. If an activity of a project fails, the project fails. The authors also consider resource constraints and the possibility of outsourcing. Modular projects, i.e., projects that include the execution of several modules, each of which consisting of several activities, have been considered in [3, 6]. For such a project to be successful every module must succeed. A module succeeds if at least one of its activities succeeds. In the former work, activity durations are deterministic and activities must be performed sequentially, while in the latter, the durations are stochastic and the resources unlimited.

Following on the work of Ranjbar and Morteza [23], we focus on a single firm facing a R&D project or the development of a new product. There are several alternatives of executing the project and its success requires the successful execution of at least one of the available alternatives. Each alternative consists of a single activity and is characterized by a cost, a duration, and a probability of technical success. The successful completion of the project provides a given payoff. These alternatives can be pursued either in parallel or sequentially. The objective is to schedule the activities in such a way as to maximize the expected Net Present Value

(eNPV) of the project. The eNPV takes into account the activity costs, the cash flows generated by the successful completion of project, the activity durations and starting times, and the probability of failure of each of the activities. Some alternatives may imply the implementation of other alternatives. This is a recently proposed problem and it has been addressed by exact methods only [22, 23]. Since this is an NP-*hard* problem (see [23]), an exact algorithm without an exponential time complexity is unlikely to exist. Thus, in here we propose a genetic algorithm since only heuristic methods are able to solve real sized problems.

Section 2 defines and provides a mathematical programming model for this problem. Section 3 discusses the methodology proposed to solve the problem and in Sect. 4 the computational experiments are reported. Finally in Sect. 5 some conclusions are drawn.

2 Problem Definition and Formulation

Given a project for which there are several alternative ways of execution, one wishes to determine the order in which these alternatives should be executed such that the project expected net present value is maximized. Alternatives pursue a similar target and consist of one activity.[1] Activities should be executed without interruption and are characterized by a cost, a duration, a set of precedence constraints, and a probability of technical success. Activity costs are to be paid at the start of the activity. The outcomes of the different tasks are considered to be independent. The successful completion of a project provides a payoff and is achieved if at least one alternative is successfully executed.

Before introducing the mathematical programming model, let us illustrate the problem by resorting to the example used in [23]. Consider a project consisting of five alternatives, for which the information is given in Table 1. Note that the execution of activity 4 requires activity 1 to be previously executed. Nevertheless, activity 4 can be executed and be successful regardless of the outcome of the execution of activity 1. It is assumed a 5 % monthly discount rate, a project deadline of 29 months, and a project payoff, achieved in case of technological success, of 2,770 dollars.

These alternatives can be scheduled in many different ways. The two extreme ones being the parallel and the sequential schedules. These schedules are given in Fig. 1.

Note that, while the parallel schedule anticipates the project completion and thus the net payoff is larger, it also leads to the highest costs since it starts activities without waiting to find out if the previously started one has had success. Thus, in this type of schedules some, in progress, alternatives of the project will be ignored.

[1] Since each alternative consists of a single activity, here and hereafter we will use indifferently alternative and activity.

Table 1 Alternatives data (the costs are given in dollars and the duration in months)

Alternative number	Costs ($)	Duration (months)	PTS	Precedent activities
1	51	8	0.73	–
2	31	6	0.62	–
3	87	3	0.91	2
4	28	7	0.57	1
5	80	4	0.86	–

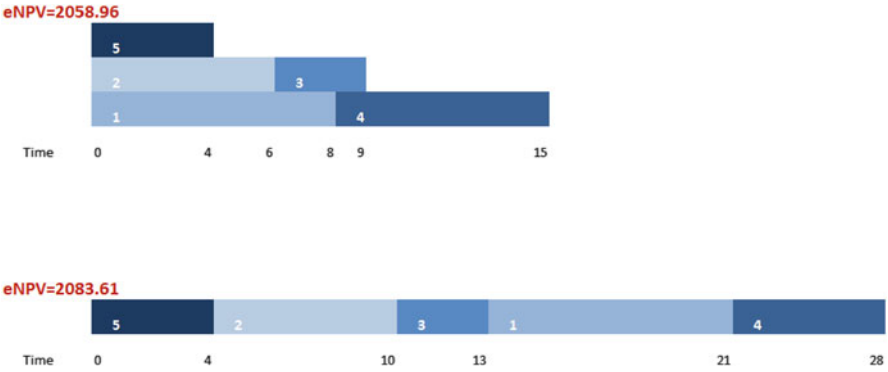


Fig. 1 Extreme schedules: parallel and serial schedules

Consider that activity 5 is successfully executed, which happens at time 4. Thus, at time 4 the project has been successfully completed, as it only requires that at least one alternative is successfully executed. Activities 1 and 2 have been initiated and paid for at time 0, and at time 4, although they are still undergoing, the outcome of their execution becomes irrelevant and they will be ignored. Here the costs are typically higher. In the serial schedule, since only one activity is being performed at any time this risk does not exist. Therefore, it is more conservative in terms of costs. However, in this case if an alternative fails it takes longer to be able to have another tried and thus, the project payoff is typically smaller since it is obtained later. Therefore, a trade-off between costs and project payoff (project duration) must be searched for.

At time t the project payoff C is obtained if and only if at least one of the activities finishing at time t (A_t) succeeds and all activities that have finished before time t (B_t) have failed; otherwise the payoff had already been received. Thus, the expected payoff at time t is given by

$$\prod_{j\in B_t}(1-p_j)\times\left(1-\prod_{k\in A_t}(1-p_k)\right)\times C. \tag{1}$$

As said before, each activity i has a cost (c_i) associated to its execution that must be paid up-front, i.e., at the time that the activity is started (s_i). In addition, an

Table 2 Notation used for the mathematical programming model

Symbols	Description
N	Set of available alternatives
i,j,k	Alternative indices
c_i	Cost of alternative i
d_i	Duration of alternative i
p_i	Probability of technical success of alternative i
$t_{\max}$	Project deadline
t	Time index
C	Project payoff
r	Discount rate
A	Set of precedence constraints
B_t	Auxiliary decision variable: set of alternatives finishing before t
A_t	Auxiliary decision variable: set of alternatives finishing at t
s_i	Decision variable: starting time of alternative i

activity is only started if all activities that have finished before (B_{s_i}) or at (A_{s_i}) its starting time have failed; otherwise the project had already been concluded. Thus, the expected cost incurred with activity i at its starting time s_i can be written as

$$\prod_{j\in\{B_{s_i}\bigcup A_{s_i}\}}(1-p_j)\times c_i. \tag{2}$$

A summary of the notation used is provided in Table 2.

The project net value is then obtained by subtracting all expected costs from all expected payoffs. However, since we are maximizing the project expected net present value, the costs and payoffs given by Eqs. (1) and (2) need to be discounted. The scheduling decisions are only constrained by the precedence relations amongst the alternatives. Therefore, the complete mathematical model is as given in Eqs. (3) to (5):

$$\text{Minimize} \quad \sum_{t=1}^{t_{max}}\left(\prod_{j\in B_t}(1-p_j)\times\left(1-\prod_{k\in A_t}(1-p_k)\right)\times C\times e^{-rt} - \prod_{j\in\{B_{s_i}\bigcup A_{s_i}\}}(1-p_j)\times c_i\times e^{-rs_i}\right) \tag{3}$$

Subject to

$$s_i+d_i\le s_j, \qquad \forall j\in A \text{ and } \forall i\in N. \tag{4}$$

$$s_i\in\mathbb{N}, \qquad \forall i\in\mathbb{N}^+. \tag{5}$$

3 Methodology

In this section, we provide an overview of the proposed solution process. This is followed by a discussion on the proposed Biased Random-Key Genetic Algorithm (BRKGA), including detailed descriptions of the solution encoding and decoding procedures, evolutionary process, and fitness function.

3.1 Overview

The new approach is based on a constructive heuristic algorithm which inserts activities, one at a time, in a partial schedule for the problem. Once all the activities are inserted, a solution is obtained. The new approach proposed in this chapter combines a BRKGA with a novel insertion decoding procedure. The role of the genetic algorithm is to evolve the encoded solutions, or *chromosomes*, which represent the *parameters* that will be used by the solution builder to construct a schedule. For each chromosome, the following phases are applied to decode the chromosome:

1. *Decoding of the parameters*: this first phase decodes the chromosome into a sequence of activities, as well as each activity scheduling mode (SM). The former determines the activities to be started, while the latter determines whether each activity is scheduled forward or backward.
2. *Construction of a solution*: The second phase makes use of the activities and SMs defined in phase 1 and uses the solution builder procedure to construct a schedule.
3. *Fitness evaluation*: The final phase computes the fitness of the solution, by computing the expected net present value as given in Eq. (3).

Figure 2 illustrates the sequence of decoding steps applied to each chromosome generated by the BRKGA. The remainder of this section describes in detail the genetic algorithm.

3.2 Biased Random-Key Genetic Algorithm

Random-key genetic algorithms (RKGAs) or genetic algorithms with random keys were introduced in [1] for solving sequencing or optimization problems whose solutions can be represented as permutations. In an RKGA, chromosomes are represented as vectors of randomly generated real numbers in the interval $[0, 1]$. A deterministic algorithm, the *decoder*, takes as input a chromosome and associates with it a solution of the combinatorial optimization problem for which an objective value or fitness value can be computed.

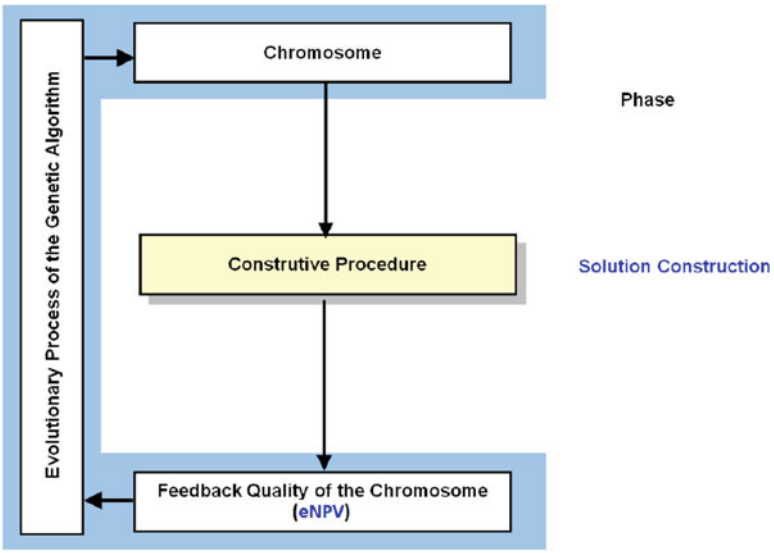


Fig. 2 Architecture of the algorithm

RKGAs are particularly attractive for sequencing problems and/or when the chromosomes have several parts (see for example [10–12, 14–17, 24], and [18]). Unlike traditional GAs, which need to use special repair procedures to handle permutations or sequences, RKGAs move all the feasibility issues into the objective evaluation procedure and guarantee that all offspring formed by crossover result into feasible solutions. When the chromosomes have several parts, traditional GAs need to use different genetic operators for each part. However, since RKGAs use the *parameterized uniform crossover* of Spears and Dejong [29] (instead of the traditional one-point or two-point crossovers), they do not need to have different genetic operators for each part.

An RKGA evolves a *population* of random-key vectors over a number of *generations* (iterations). The initial population is made up of p vectors of r random keys. Each component of the solution vector, or random key, is generated independently at random in the real interval $[0, 1]$. After the fitness of each individual is computed by the decoder in generation g, the population is partitioned into two groups of individuals: a small group of p_e *elite* individuals, i.e., those with the best fitness values, and the remaining set of $p - p_e$ *non-elite* individuals. To evolve a population g, a new generation of individuals is produced. All elite individuals of the population of generation g are copied without modification to the population of generation $g + 1$. RKGAs implement mutation by introducing *mutants* into the population. A mutant is a vector of random keys generated in the same way that an element of the initial population is generated. At each generation, a small number p_m of mutants is introduced into the population. With $p_e + p_m$ individuals accounted for in population $g + 1$, $p - p_e - p_m$ additional individuals need to be generated to

complete the p individuals that make up population $g+1$. This is done by producing $p-p_e-p_m$ offspring solutions through the process of *mating* or *crossover*.

A BRKGA [13] differs from an RKGA in the way parents are selected for mating. While in the RKGA of Bean [1] both parents are selected at random from the entire current population, in a BRKGA each element is generated by combining a parent selected at random from the elite partition of the current population with another from the rest of the population, also randomly selected. Repetition in the selection of a parent is allowed and therefore an individual can produce more than one offspring in the same generation. As in RKGAs, parameterized uniform crossover is used to implement mating in BRKGAs. Let ρ_e be the probability that the vector component of an elite parent is inherited by the offspring. For $i = 1,\ldots,r$, the ith component $c(i)$ of the offspring vector c takes on the value of the ith component $e(i)$ of the elite parent e with probability ρ_e and the value of the ith component $\bar{e}(i)$ of the non-elite parent $\bar{e}$ with probability $1-\rho_e$.

Once the next population is complete, the corresponding fitness values are computed for all of the newly created random-key vectors and the population is partitioned into elite and non-elite individuals to start a new generation.

A BRKGA searches the solution space of the combinatorial optimization problem indirectly by searching the r-dimensional continuous hypercube, using the decoder to map solutions in the hypercube to solutions in the solution space of the combinatorial optimization problem where the fitness is evaluated.

To specify a BRKGA, one simply needs to specify how solutions are encoded and decoded and how their corresponding fitness values are computed. This is done in the next sections.

3.2.1 Chromosome Representation and Decoding

A chromosome encodes a solution to the problem as a vector of random keys. In a direct representation, a chromosome represents a solution to the original problem and is called *genotype*, while in an indirect representation, it does not and special procedures are needed to obtain from it a solution called a *phenotype*. In the present context, the solutions will be represented indirectly by parameters that are later used by a decoding procedure to obtain a solution. To obtain the solution (phenotype) we use the decoding procedures described in Sect. 3.2.2.

In this chapter, a solution to the problem is represented indirectly by the chromosome structure given in Fig. 3, where n is the number of activities. Overall the chromosome has $n+(n-1)2n$ genes.

The genes in blue are used by the solution builder (decoding procedure) to determine which activity or activities are to be scheduled at each iteration into the partial schedule and the genes in red are used to decide whether the activity chosen is going to be scheduled forward or backward. Note that in the first iteration the activities must always be scheduled forward. An activity is considered schedulable if all of its predecessors have already been scheduled and if its blue gene value is

$$\text{Chromosome} = (\text{gene}_1, \ldots, \text{gene}_n, \text{gene}_{n+1}, \ldots, \text{gene}_{n+n}, \ldots, \text{gene}_{n+(n-2)2n+1}, \ldots, \text{gene}_{n+(n-2)2n+n}, \text{gene}_{n+(n-2)2n+n+1}, \text{gene}_{n+(n-2)2n+n+n})$$

Fig. 3 Chromosome representation

1	2		3		4		5	
Act	Act	S/E	Act	S/E	Act	S/E	Act	S/E
0.7	0.2	0.4	0.12	0.41	0.12	0.41	0.4	0.23
0.2	0.1	0.3	**0.9**	**0.3**	0.09	0.03	0.55	0.58
0.1	0.33	0.3	0.23	0.3	**0.63**	**0.37**	0.6	0.55
0.25	0.46	0.2	0.26	0.2	**0.76**	**0.14**	0.5	0.95
0.14	**0.85**	**0.62**	0.15	0.62	0.05	0.62	0.4	0.12

Fig. 4 Chromosome example used in the illustration of the solution builder

greater than or equal to 0.5. If the value of the red gene is greater than or equal to 0.5 then the chosen activity is scheduled backward; otherwise it is scheduled forward.

The decoding (mapping) of each chromosome into a schedule is performed by the solution builder, which is described in the next section.

3.2.2 Solution Builder

The solution builder follows a sequential process that inserts activities into a partial schedule. The order in which activities are inserted into the partial schedule and the corresponding mode (forward or backward) are evolved by the BRKGA. The solution builder comprises the following two main steps:

1. Selection of activities to be inserted;
2. Selection of the mode used for the insertion in the partial solution of the activities selected in step 1.

The possible insertion times for scheduling an activity are provided by the starting (S) or ending times (E) of the activities already scheduled. Amongst these, we are only interested on the ones that are feasible regarding the precedence relations between activities.

To illustrate how the solution builder works we used again the example provided in Table 1. A solution will be constructed using the chromosome in Fig. 4.

Initially only time zero is available for scheduling one or more activities. According to the precedence constraints the activities which are schedulable are 1, 2, and 5. However, only activity 1 has a blue gene value greater than or equal to

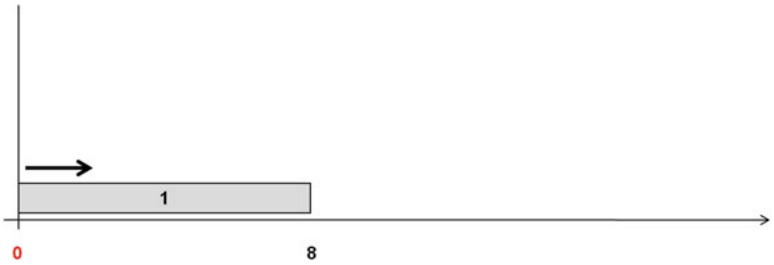


Fig. 5 Step 1 of solution builder: partial schedule after inserting activity 1

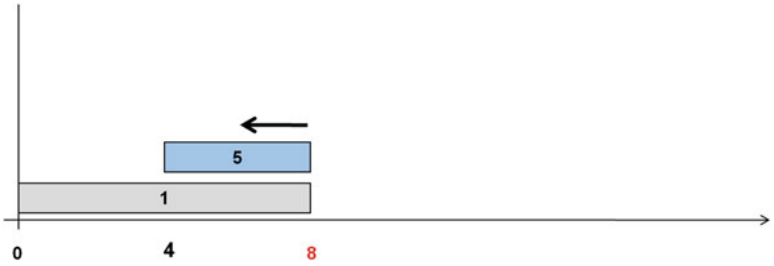


Fig. 6 Step 2 of solution builder: partial schedule after inserting activity 5

0.5 (0.7), so activity 1 is the only one selected for insertion into the partial schedule. Since this activity is scheduled at time 0, it must be scheduled forward. At this point the partial schedule looks like the one given in Fig. 5.

The next time to be considered for insertion is time 8. Only activities 2, 4, and 5 can be considered since due to the precedence constraints activity 3 cannot yet be scheduled. According to the second column only activity 5 has a blue gene value greater than or equal to 0.5 (0.85), thus only activity 5 can be scheduled. Given that the value in sub-column S/E of column 2 for activity 5 is 0.62 (>0.5), then activity 5 is scheduled backward. At this point the partial schedule looks like the one given in Fig. 6.

The next time to be considered for insertion is time 4, the only one available not yet considered. According to the precedence constraints only activity 2 can be started and its blue gene has a value greater than or equal to 0.5 (0.9), thus activity 2 is inserted into the partial schedule. Since the value in sub-column S/E of column 3 for activity 2 is smaller than 0.5 (0.3), then activity 2 is scheduled forward. The partial schedule obtained is illustrated in Fig. 7.

The next time to be considered for insertion is time 10. According to the precedence constraints and the fourth column of the chromosome, the activities which can be started are activities 3 and 4. Since the value for both in sub-column S/E is smaller than 0.5 (0.37 and 0.14, respectively), both are scheduled forward. Given that there are no more activities to be scheduled the final schedule is the one given in Fig. 8.

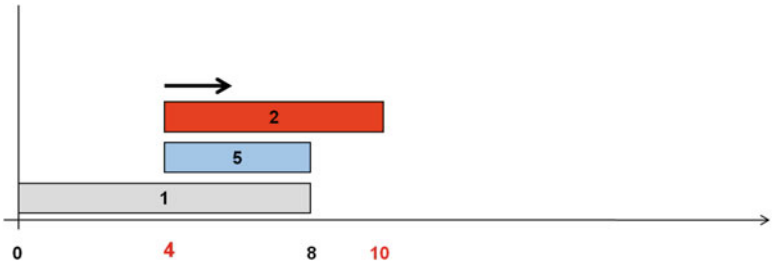


Fig. 7 Step 3 of solution builder: partial schedule after inserting activity 2

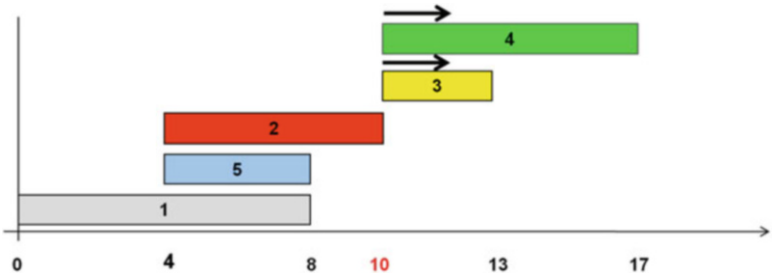


Fig. 8 Step 4 of solution builder: final schedule after inserting activities 3 and 4

Finally, the fitness of the solution, i.e., the project expected net present value, is computed according to Eq. (3) and in this case it is 1,702.87 dollars.

4 Computational Experiments

The methodology proposed here was tested on the randomly generated test problem instances used by Ranjbar and Morteza [23].

The 60 problem instances used have been generated using RanGen [9]. Four different problem sizes and three different order strength values[2] have been considered. For each of these 12 combinations five problem instances were generated by choosing uniform random values for durations, costs, and PTS in the intervals [1,10], [10,100], and [0.5,1], respectively. For each problem instance the payoff has been chosen to be five times the sum of the alternatives cost and the discount rate was set to 5 %.

[2]The number of precedence-related activity pairs divided by the theoretically maximum number of such pairs in the network [20].

The genetic algorithm has been coded using C++ and the experiments have been carried out on a computer with an Intel Core i7-2630QM @2.0 GHz CPU running the Linux operating system with Fedora release 16.

We compare the best solutions obtained with the genetic algorithm with those of the branch and bound developed by Ranjbar and Morteza [23]. The BRKGA was able to find an optimal solution to 47 of the 60 problem instances considered. The computational time required by the BRKGA was always below 10 s and on average was about 7 s.

The BRKGA proposed here, when compared to the best alternative method [23], provides an enormous improvement since it improves substantially the computational time performance. In addition, it finds very good solutions, actually optimal for most problems. It should be noticed that for the worst case class of problems (problems with 12 alternatives and order strength of 0.4), the alternative method takes around 1 h and 45 min. The optimality gaps are always below 2.5 % and the average optimality gap for the 60 problems solved is below 0.2 %.

5 Conclusions

We have presented a genetic algorithm for scheduling projects with alternative tasks subject to technical failure. This is a newly proposed problem and thus far only branch-and-bound algorithms have been proposed. Results obtained compare favorably with the ones reported in current literature.

The genetic algorithm proposed finds nearly optimal solutions, actually optimal for most solved problem instances. The idea is that it improves for all cases. However, the improvement is praticularly relevant for larger size problem instances. The magnitude of the improvement grows with problem size. For the 12 alternative problems with order strength of 0.4 the BRKGA requires less than 10 s, while the literature reports about 1 h and 44 min. Nevertheless, the average gap is only about 0.2 %.

Acknowledgements This work was partially supported by projects PTDC/EGE-GES/117692/2010 and NORTE-07-0124-FEDER-000057 funded by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF) and the Programme COMPETE, and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

1. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comput. **6**, 154–160 (1994)
2. Colvin, M., Maravelias, C.T.: R&d pipeline management: task interdependencies and risk management. Eur. J. Oper. Res. **215**(3), 616–628 (2011)

3. Coolen, K., Wei, W., Nobibon, F.T., Leus, R.: Scheduling modular projects on a bottleneck resource. J. Sched. **17**(1), 67–85 (2014)
4. Creemers, S., Leus, R., De Reyck, B., Lambrecht, M.: Project scheduling for maximum npv with variable activity durations and uncertain activity outcomes. In: IEEE International Conference on Industrial Engineering and Engineering Management, 2008. IEEM 2008, pp. 183–187, 2008
5. Creemers, S., De Reyck, B., Leus, R.: R&d project planning with multiple trials in uncertain environments. In: IEEE International Conference on Industrial Engineering and Engineering Management, 2009. IEEM 2009, pp. 325–329, 2009
6. Creemers, S., De Reyck, B., Leus, R.: Project planning with alternative technologies in uncertain environments. FEB Research Report KBI_1314, 2013
7. De Reyck, B., Leus, R.: R&d project scheduling when activities may fail. IIE Trans. **40**(4), 367–384 (2008)
8. De Reyck, B., Grushka-Cockayne, Y., Leus, R.: A new challenge in project scheduling: the incorporation of activity failures. Tijdschrift voor economie en management **52**(3), 411 (2007)
9. Demeulemeester, E., Vanhoucke, M., Herroelen, W.: Rangen: a random network generator for activity-on-the-node networks. J. Sched. **6**(1), 17–38 (2003)
10. Fontes, D.B.M.M., Gonçalves, J.F.: A multi-population hybrid biased random key genetic algorithm for hop-constrained trees in nonlinear cost flow networks. Optim. Lett. **7**(6), 1303–1324 (2013)
11. Gonçalves, J.F., Almeida, J.R.: A hybrid genetic algorithm for assembly line balancing. J. Heuristics **8**, 629–642 (2002)
12. Gonçalves, J.F., Resende, M.G.C.: An evolutionary algorithm for manufacturing cell formation. Comput. Ind. Eng. **47**, 247–273 (2004)
13. Gonçalves, J.F., Resende, M.G.C.: Biased random-key genetic algorithms for combinatorial optimization. J. Heuristics **17**, 487–525 (2011)
14. Gonçalves, J.F., Resende, M.G.C.: A parallel multi-population biased random-key genetic algorithm for a container loading problem. Comput. Oper. Res. **39**(2), 179–190 (2012)
15. Gonçalves, J.F., Resende, M.G.C.: A biased random key genetic algorithm for 2d and 3d bin packing problems. Int. J. Prod. Econ. **145**(2), 500–510 (2013)
16. Gonçalves, J.F., Mendes, J.J.M., Resende, M.G.C.: A hybrid genetic algorithm for the job shop scheduling problem. Eur. J. Oper. Res. **167**, 77–95 (2005)
17. Gonçalves, J.F., Mendes, J.J.M., Resende, M.G.C.: A genetic algorithm for the resource constrained multi-project scheduling problem. Eur. J. Oper. Res. **189**, 1171–1190 (2009)
18. Gonçalves, J.F., Costa, M.D., Resende, M.G.C.: A biased random-key genetic algorithm for the minimization of open stacks problem. Int. Trans. Oper. Res. (2014, to appear) DOI: 10.1111/itor.12109
19. Maravelias, C.T., Grossmann, I.E.: Simultaneous planning for new product development and batch manufacturing facilities. Ind. Eng. Chem. Res. **40**(26), 6147–6164 (2001)
20. Mastor, A.A.: An experimental investigation and comparative evaluation of production line balancing techniques. Manag. Sci. **16**(11), 728–746 (1970)
21. Miguel, J.L., Schaefer, E., Reklaitis, G.V.: Challenges and opportunities in enterprise-wide optimization in the pharmaceutical industry. Comput. Chem. Eng. **47**, 19–28 (2012)
22. Ranjbar, M.: A branch-and-bound algorithm for scheduling of new product development projects. Int. Trans. Oper. Res. **20**(2), 251–266 (2013)
23. Ranjbar, M., Davari, M.: An exact method for scheduling of the alternative technologies in r&d projects. Comput. Oper. Res. **40**(1), 395–405 (2013)
24. Roque, L.A.C., Fontes, D.B.M.M., Fontes, F.A.C.C.: A hybrid biased random key genetic algorithm approach for the unit commitment problem. J. Comb. Opt. **28**(1), 140–166 (2014)
25. Schmidt, C.W., Grossmann, I.E.: Optimization models for the scheduling of testing tasks in new product development. Ind. Eng. Chem. Res. **35**(10), 3498–3510 (1996)
26. Schmidt, C.W., Grossmann, I.E., Blau, G.E.: Optimization of industrial scale scheduling problems in new product development. Comput. Chem. Eng. **22**, S1027–S1030 (1998)

27. Sobek, D.K., Ward, A.C., Liker, J.K.: Toyota's principles of set-based concurrent engineering. Sloan Manag. Rev. **40**(2), 67–84 (1999)
28. Sommer, S.C., Loch, C.H.: Selectionism and learning in projects with complexity and unforeseeable uncertainty. Manag. Sci. **50**(10), 1334–1347 (2004)
29. Spears, W.M., Dejong, K.A.: On the virtues of parameterized uniform crossover. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 230–236, 1991

Discrete Competitive Facility Location: Modeling and Optimization Approaches

Athanasia Karakitsiou

Abstract Competitive facility location problems are concerned with the following situation: a firm wants to locate a predefined number of facilities to serve customers located in some region where there already exist (or will be) other firms offering the same service. Both new and existing firms compete for optimizing their market share of profit. A discrete version of such problems arises when it is assumed that there are a (rather small) finite number of candidate locations and the markets consist of point demands. We review modeling and optimization approaches for this type of problems and we emphasize and develop the bi-level programming methodology.

Keywords Competitive facility location • Bi-level programming • $(r|p)$-Centroid problem • $(r|X_p)$-Medianoid problem • Location under customers competition

1 Introduction

Facility location analysis is one of the most well-studied areas of the operations research. In the basic model, there is a predefined cost for opening a facility and also connecting a customer to a facility, the goal is to minimize the total cost.

The typical facility location problem assumes that the locating facility is either a price taker or a monopolist, so that the market competition is neglected among the companies. However this simplified assumption does not fit in most real-life situations and the need arises to incorporate competition among the decision-makers. Indeed, competitive location models additionally incorporate the fact that location decisions have been or will be made by independent decision-makers who will subsequently compete with one another for market share, profit maximization, etc. In addition, the assignment of customers being served by these facilities and how these facilities are connected with each other are interesting decisions considered within the problem.

A. Karakitsiou (✉)
ETS Institute, Industrial Logistics, Luleå University of Technology, 971 87 Luleå, Sweden
e-mail: athkar@ltu.se

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_8

It is widely accepted that the competitive location analysis was initiated by Hotelling [14]. In his two ice cream vendors game, he examined location policies of an interdependently acting duopoly in a linear market of a given length. The distribution of buying power along the line segment is assumed uniform. Each customer has an inelastic demand for the good and pays the transportation cost of obtaining the good. Therefore, he patronizes the nearest facility in order to minimize his expenditures. He proved that a "back to back" location in the middle of the market constitutes a long-run equilibrium. Since then, a vast number of publications have been devoted to the subject. Thus, different classification efforts with respect to multiple components have been proposed in the literature, see for example, [9, 10, 19, 27] among others. Spatial representation and the nature of competition are some of them.

The classification based on the spatial representation classifies the CFL models into three broad categories: (a) *Continuous models*, where the potential location of the facilities can be anywhere in the plane, (b) *Discrete models*, where facilities are allowed to locate at a finite set of possible locations, and (c) *network models*, where any point on a network is suitable for location. From the optimization point of view, the techniques used to cope with the problems also differ. Continuous location problems are, for most of the cases, nonlinear optimization problems, while discrete and network location problems are integer programming/combinatorial optimization problems.

When the nature of competition is used as a classification method then again three different classes of problems can be identified. (a) *Static problems,* which assume that a firm enters into a market, where operate same existing firms, aiming at choosing the location of p facilities such as to attract the maximum market share. The new competitor enters into the market by having full and in advance information about the characteristics and the strategies of the existing firm (s). It is assumed further that this information is fixed and no reaction is expected from the existing competitor(s). When the assumption of the non reacting competitors is eliminated, two new classes of CLF arise, (b) *dynamic* and (c) *sequential* location problems. The competing firms make a location decision simultaneously in the first case, whereas there is a hierarchy in the decision-making process in the second. The sequential location completion is mainly formulated as a Stackelberg-type game. On the other hand, in simultaneous location games, the Nash equilibrium constitutes the solution of the problem.

In this work we focus on discrete bi-level CFL problems. Our aim is to provide an up to date review of modeling and optimization approaches used in the bibliography. Moreover, we develop a new bi-level methodology for this type of problem.

2 Sequential Deterministic Facility Location Problems

The formalization of this class of problems and fundamental complexity results were established by Hakimi [11]. Following the game introduced by von Stackelberg [30] Hakimi [11] presented the two basic problems in the sequential location

analysis, the *centroid* and *medianoid* problems. These two problems are faced by the leader and the follower, respectively. The leader attempts to locate $p \geq 1$ facilities knowing that the follower will in turn locate his $r \geq 1$ facilities based on the leader's chosen location; this is the $(r|p)$-centroid problem. The follower knows the set X_p that indicates where the leader's facilities are located and solves an $(r|X_p)$-medianoid problem. Customers choose among the facilities according to a function of the distance between them and the facilities, preferring always the closest. This is the so-called binary customer choice. The formulation of the problems is based on the assumption that co-location is not allowed and if, by any chance, the distance from a customer to the closest facility of the two competitors is the same, the customer always prefers the leader's facility. The demand of the customer is also considered to be inelastic with respect to distance travelled.

Given the set I of m potential facilities location and J the set of n customers locations, let x_{ij} defines the distance between customer j and facility i. It is assumed further that w_j is the weight (profit, demand, etc.) of customer j.

If X and Y denote the location occupied by the leader and the follower, respectively, and $d(j,X)$ and $d(j,Y)$ the distance between customer j and his nearest facility from X and Y, respectively, then customer j will prefer Y over X if $d(j,Y) < d(j,X)$ and he prefers X over Y otherwise. If $J(Y \prec X)$ is the set of customers who prefer Y over X then $W(Y \prec X) = \sum_{j \in J(Y \prec X)} w_j$ denotes the total weight of the customers who prefer Y over X.

For each X the follower's strategy is the set of other location Y that provides the maximal market share, $W^*(X)$, to him. This maximal market share is obtained by solving the following problem:

$$\max_{Y,|Y|=r} W(Y \prec X). \tag{1}$$

The leader on the other hand is interested in maximizing his own market share. Thus, his optimal location strategy X^* is the one that minimizes the follower's market share. Therefore, the leader's maximal market share is obtained by solving the following problem:

$$\min_{X,|X|=p} \max_{Y,|Y|=r} W(Y \prec X). \tag{2}$$

Hakimi [12] extended the initial formulation of the problem by considering elastic demand and different customer choice rules, apart from the binary choice rule, such as partially binary choice and the proportional preference choice of the customers. Under the partially binary choice the customer uses the closest facility of each firm. Under the proportional choice the customer proportionally distributes his demand among the operating facilities. He came up with six different scenarios and he stated several vertex optimality results. Particularly, he proved the existence of a nodal solution for the partially binary problem, under both inelastic and elastic demands. He proved also that a nodal solution exists in the proportional choice-

elastic demand case only if the demand captured by the facilities is a linear function of the distance. Suárez-Vega et al. [28] extended his result by considering a concave function of the demand capture.

A 3-level formulation for both leader's and follower's problem and a heuristic solution procedure based on the elimination procedure in a candidate list are proposed in [6]. They formulated the problem as a three-stage optimization process which included the customer selection problem, the follower location problem, and the leader location problem. The corresponding problem, $(r|p)$-centroid problem, with inelastic demand is as follows:

$$\max \sum_{i=1}^{m} \left[\sum_{k=1}^{n} h_k z_{ki} \right] x_i \tag{3}$$

$$s.t \quad \sum_{i=1}^{m} x_i = p \tag{4}$$

$$x_i \in \{0,1\}, i \in [1,\ldots,m] \tag{5}$$

where z solves

$$CUS(x,y) \ \min \sum_{k=1}^{n} \sum_{i=1}^{m} d_{ki} h_k z_{ki} \tag{6}$$

$$st \ \sum_{i=1}^{m} z_{ki} = 1, k \in [1,\ldots,n] \tag{7}$$

$$z_{ki} \leq \bar{x_i} + \bar{y_i}, k \in [1,\ldots,n], i \in [1,\ldots,m] \tag{8}$$

$$z_{ki} \in \{0,1\}, k \in [1,\ldots,n], i \in [1,\ldots,m] \tag{9}$$

where y solves

$$FLO_r \ \max \sum_{i=1}^{m} \sum_{k=1}^{n} h_k z_{ik} \tag{10}$$

$$st \ \sum_{i=1}^{m} y_i = r, \tag{11}$$

$$\sum_{i=1}^{m} z_{ki} \leq 1, k \in [1,\ldots,n] \tag{12}$$

$$z_{ki} - c_{ki} y_i \leq 0, k \in [1,\ldots,n], i \in [1,\ldots,m] \tag{13}$$

$$z_{ki}, y_i \in \{0,1\}, k \in [1,\ldots,n], i \in [1,\ldots,m] \tag{14}$$

In the above model m is the number of possible facility locations and n is the number of customer locations. $d_{ki} = d(c_k, f_i)$ is the distance between the kth customer location c_k and the ith facility point f_i. h_k is the total demand of the

customers located at c_k. A set Z of location points is identified by a binary vector $z = (z_i : i \in [1,\ldots,m])$ where $z_i = 1$ if $f_i \in Z$ and $z_i = 0$ if $f_i \notin Z$. The decision variables in the leader and follower location problems are the m-vectors x and y of $0-1$ or binary decision variables corresponding to sets X and Y. z_{ki} is the 0–1 decision variables indicating whether the customers located at the k customer location c_k prefer the location f_i for the facility and $c_{ki} = 1$ if $d_{ki} < \min\{d_{kj}\bar{x}_j = 1\}$. The $CUS(x,y)$ is the customer selection problem. The objective function of this problem represents the total distance travelled by the customers to arrive at the corresponding facility points. The constraints state that each customer has to go to one location in the leader location set or in the follower location set. FLO_r corresponds to the follower's location problem. Campos Rodríguez et al. [6], based on the observation that the mathematical programming formulation of the minimax problem that corresponds the leader's problem (3)–(14) is

$$\max \quad W \tag{15}$$

$$st \quad |x| = p \tag{16}$$

$$W(Y \prec X) \leq W, \forall Y \in L^r, \tag{17}$$

proposed a heuristic based on an elimination procedure in a candidate list in order to solve the leaders problem. In the procedure, a leader solution provides an upper bound for the leader follower problem. A family F of good follower candidates is used to conclude that the upper bound provided by a leader solution cannot be improved, and therefore, this solution is an optimal solution.

The bi-level formulation of Hakimi's model proposed by Alekseeva et al. [1] employs three kinds of binary variables:

$$x_i = \begin{cases} 1 \text{ if facility } i \text{ is opened by leader,} \\ 0 \text{ otherwise,} \end{cases} \tag{18}$$

$$y_i = \begin{cases} 1 \text{ if facility } i \text{ is opened by follower,} \\ 0 \text{ otherwise,} \end{cases} \tag{19}$$

$$z_j = \begin{cases} 1 \text{ if customer } j \text{ is served by leader,} \\ 0 \text{ otherwise,} \end{cases} \tag{20}$$

It assumes also that for a given solution, x, used by the leader, the set

$$J_j(x) = \{i \in I | d_{ij} < \min_{l \in I | x_i = 1} d_{lj}\}, j \in J$$

defines the set of facilities which allows the follower to capture customer j.

$$\max_x \sum_{i \in J} w_j z_j^*(x) \tag{21}$$

$$st. \quad \sum_{i \in I} x_i = p, \tag{22}$$

$$x_i \in \{0,1\}, \forall i \in I \tag{23}$$

where $z_j^*(x)$ solves

$$\max_{y,z} \sum_{i \in J} w_j (1 - z_j) \tag{24}$$

$$st. \quad \sum_{i \in I} y_i = r \tag{25}$$

$$1 - z_j \leq \sum_{i \in I_j(x)} y_i, j \in J, \tag{26}$$

$$x_i + y_i \leq 1, i \in I \tag{27}$$

$$y_i, z_i \in \{0,1\}, i \in I, j \in J. \tag{28}$$

Thereafter, a hybrid memetic algorithm is used for the solution of the problem. The improvement of the elements of population at each iteration is done through a probabilistic Tabu search procedure.

An upper bound is obtained by reformulating the bi-level problem as a single level mixed integer problem with an exponential number of constraints and variables. If $\mathcal{F}$ is a family of follower solutions and $I_j(y) = \{i \in I | d_i j \leq \min_{l \in I}(d_{lj} | y_l = 1)\}$, $y \in \mathcal{F}, j \in J$ is the set of facilities which allow the leader to keep the customer j if the follower uses solution y, and if $\mathcal{F}$ contains all possible solutions of the follower then problem (21)–(28) is equivalent to the following $0-1$ program:

$$\max \quad W \tag{29}$$

$$st \tag{30}$$

$$\sum_{j \in J} w_j x_{iy} \geq W, y \in \mathcal{F}, \tag{31}$$

$$z_{iy} \leq \sum_{i \in J_j(y)} x_i, j \in J y \in \mathcal{F}, \tag{32}$$

$$\sum_{i \in I} x_i = p, \tag{33}$$

$$x_i, z_{iy} \in \{0,1\}, i \in I, j \in J, y \in \mathcal{F} \tag{34}$$

where $W \geq 0$ is the total market share of the leader and z_{iy} is binary variable indicating whether customer j is serviced by the leader when the follower uses a solution y.

This single level model is also used to find the global optimum. An iterative exact algorithm is developed for this purpose. Alekseeva et al. [2] have used the single level formulation proposed by [26] in order to improve exact iterative method previously developed in [1].

The authors in [7], based on the bi-level representation (21)–(28), approached leader's–follower's problem using two metaheuristics methods: local search with variable neighborhoods and stochastic Tabu search.

The bi-level models proposed by Beresnev [3] contain the fixed cost for opening facilities. The author considers two settings of the problem that differ in the objective functions of the follower firm: In the first, it is assumed that the goal of the leader firm as well as the follower firm is the maximization of the profit, while in the second, the objective of the follower is maximization of his income. It is also assumed that each facility opened by the follower firm cannot be loss-making. The author uses the following notation in order to build up the proposed models

- $I = \{1,\ldots,m\}$ is the set of possible sites for location;
- $J = \{1,\ldots,n\}$ is the set of clients;
- p_{ij} is the profit realized by facility $i \in I$ opened by the leader when serving client $j \in J$
- $\prec_j$ is a linear order on I determining the preferences of client $j \in J$, and $i \prec_j k$ means that of the two open facilities i and $k \in I$ client j selects facility i; the relation $i \preceq_j k$ means that either $i \prec_j k$ or $i = k$;
- f_i is the fixed cost of the leader firm for opening facility $i \in I$;
- g_i is the fixed cost of the follower firm for opening facility $i \in I$.
- x_i is the variable indicating if facility $i \in I$ is opened by the leader firm,
- x_{ij} is the variable indicating if facility $i \in I$ opened by the leader firm is selected by client $j \in J$;
- z_i is the variable indicating if the follower firm opens facility $i \in I$;
- z_{ij} is the variable indicating if client $j \in J$ selects facility $i \in I$ opened by the follower firm

When the goal of the follower firm is to maximize the profit, it is written as follows:

$$\max_{(x_i),(x_{ij})} \left\{ -\sum_{i\in I} f_i x_i + \sum_{j\in J} \left(\sum_{i\in I} p_{ij} x_{ij} \right) \left(1 - \sum_{i\in I} \tilde{z}_{ij} \right) \right\} \tag{35}$$

$$st \qquad \sum_{i\in I} x_{ij} = 1, j \in J \tag{36}$$

$$x_i \geq x_{ij}, i \in I, j \in J, \tag{37}$$

$$x_i + \sum_{i \prec_j l} x_{lj} \leq 1, i \in I, j \in J \tag{38}$$

$$x_i, x_{ij} \in \{0,1\}, i \in I, j \in J \tag{39}$$

$(\tilde{z}_i), (\tilde{z}_{ij})$ is the optimal solution of the problem

$$\max_{(z_i),(z_{ij})} \left\{ -g_i z_i + \sum_{j\in J} \sum_{i\in I} q_{ij} z_{ij} \right\} \tag{40}$$

$$st \qquad \sum_{i \in I} z_{ij} \leq 1, j \in J \tag{41}$$

$$z_i \geq z_{ij}, i \in I, j \in J \tag{42}$$

$$x_i + z_i + \sum_{i \prec_j l} z_{lj} \leq 1, i \in I, j \in J \tag{43}$$

$$z_i, z_{ij} \in \{0,1\} i \in I, j \in J \tag{44}$$

Objective function (35) shows the value of profit received by the leader taking into account that a part of his consumers will be captured by the follower. Constraint (36) guarantees that each client can select one facility from the leader and inequalities (37) that only one open facility can be selected. Inequalities (38) implement the rule for choosing a facility opened by the leader to service a consumer. The same inequalities guarantee that to service each consumer one can choose only one facility opened by the Leader. Objective function (40) of problem shows the value of the profit received by the follower. Inequalities (43) implement conditions for the follower capturing consumers for given facilities opened by the Leader.

The computational complexity of problem (35)–(44) is discussed in [23] where the author proved that the problem is Σ_2^p-hard when the cost of opening facilities are considered null.

In a series of publication [3–5, 24], a number of solution methods of the problem have been proposed. Their main characteristic is that they are based on the maximization of a pseudo-Boolean function of the form

$$\max_x \quad f(x) \tag{45}$$

$$st \; x \in B^m \tag{46}$$

3 Sequential Probabilistic Competitive Facility Location Models

Models presented in the previous sections assume that the distance traveled is the only criterion affecting the patronizing behavior of the customers. However, in more realistic situations, customers consider other attributes of the facilities during their decision-making process such as size, quality of product, and service provided.

Huff [15] suggested to measure the attraction felt by a customer for a facility as a measure of his patronizing probability. In his model the attraction felt by a customer at zone i towards a facility j located at place x_j is proportional to the size of the facility and inversely proportional to a power of the distance between zone i and x_j. A general formulation of the attraction function is given by

$$u_{ij} = \frac{A_j}{f(d_{ij})} \tag{47}$$

where A_j is the attractiveness or quality of the facility j and f is a non-decreasing function of distance.

In the multiplicative competitive interaction (MCI) model of Nakanishi and Cooper [25] different attributes of the facility were used together by taking their product after weighting them by raising each to a power:

$$u_{ij} = \prod_{k=1}^{s} x_{ijk}^{\beta_k} \tag{48}$$

where s is the set of facility's attributes, x_{ijk} is the kth attribute describing a facility j by customers at i, and β_k is the weight of the kth attribute.

The additive utility function is utilized in [8]. A general form of this function can be

$$U = \sum_{k=1}^{s} \beta_k f_k(x_k) \tag{49}$$

where x_k is the kth attribute and β_k its associated weight.

Other models (see for example [13]) make use of the exponential attraction function which is generally given by

$$A_{ij} = a_j^{\alpha} e^{-\beta d_{ij}} \tag{50}$$

where a_j measures the quality of the facility j and α, β are parameters determined empirically.

The aim of the model proposed in [21] is to determine the optimal location and the attractiveness of the new facilities to be opened by a firm in a market where there are r existing facilities that belong to a competitor or several competitors. The goal is the maximization of the firm's profit. The customers are aggregated at $N = 1, \ldots, n$ demand points and the number of candidate facility site is $M = 1, \ldots, m$. The parameters of the problem are

- a_j annual buying power at point j
- c_i unit attractiveness cost at site i
- f_i annualized fixed cost of opening and operating a facility at i
- d_{ij} Euclidean distance between site i and point j
- b_j total utility of the existing facility depending on its attractiveness and distance from point j
- u_i maximum attractiveness level of facility to be opened at site i
- q_k attractiveness of existing facility j

and variables

- Q_i attractiveness of the facility opened at site i
- X_i binary variable that is equal to 1 if a facility is opened at site i and 0 otherwise.

By using Huff's model the utility of a facility opened at site i with attractiveness Q_i is defined by Q_i/d_{ij}^2. By using the same rule the total utility felt by customers at j for the existing facilities is $b_j = \sum_{k=1}^{r} q_k/d_{kj}^2$, where d_{kj} is the distance between demand point j and existing facility k. Hence, the market share of the facility at i is expressed as

$$P_{ij} = \frac{Q_i/d_{ij}^2}{\sum_{i=1}^{m}(Q_i/d_{ij}^2) + \sum_{k=1}^{r} q_k/d_{kj}^2} \tag{51}$$

As a result the total revenue captured by the new facility is given by

$$\sum_{j=1}^{n} a_j \frac{\sum_{i=1}^{m}(Q_i/d_{ij}^2)}{\sum_{i=1}^{m}(Q_i/d_{ij}^2) + \sum_{k=1}^{r}(q_k/d_{kj}^2)} \tag{52}$$

Then the problem can be formulated as

$$\max_{\mathbf{Q},\mathbf{X}} z = \sum_{j=1}^{n} a_j \frac{\sum_{i=1}^{m}(Q_i/d_{ij}^2)}{\sum_{i=1}^{m}(Q_i/d_{ij}^2) + \sum_{k=1}^{r}(q_k/d_{kj}^2)} - \sum_{i=1}^{m} f_i X_i - \sum_{i=1}^{m} c_i Q_1 \tag{53}$$

$$s.t\ Q_i \le u_i X_i, i = 1, \ldots, m \tag{54}$$

$$X_i \in \{0,1\}, i = 1, \ldots, m \tag{55}$$

$$Q_i \ge 0, i = 1, \ldots, m \tag{56}$$

To solve the problem three solution methods are presented. One is a heuristic based on the Lagrangian relaxation of the model, while the other two are exact procedures based on the branch and bound technique.

The model proposed in [20] allows the competitor to react in every location decision made by the firm by adjusting the attractiveness level of his own existing facilities with the objective to maximize his profit. The resulting formulation is a bi-level programming model where the entering firm is consider as the leader and the existing competitor as the follower. In this bi-level formulation, the attractiveness level at the competitor's facility q_k becomes the decision variable of the follower.

Thus, the leader solves problem (53)–(56), while the follower the problem

$$\max_{\mathbf{q}} \sum_{j=1}^{n} a_j \frac{\sum_{i=1}^{r}(q_k/d_{kj}^2)}{\sum_{i=1}^{m}(Q_i/d_{ij}^2)+\sum_{k=1}^{r}(q_k/d_{kj}^2)} - \\ - \sum_{k=1}^{r} \tilde{c}_k(q_k - \tilde{q}_k), \tag{57}$$

$$s.t. \quad q_k \leq \bar{q}_k, k = 1, \ldots, r \tag{58}$$

$$q_k \leq 0, k = 1, \ldots, r \tag{59}$$

where the first term of the objective function represents the follower's market share, and $\tilde{q}_k, \bar{q}_k, \tilde{c}_k$ are parameters representing the current attractiveness level, the maximum attractiveness level, and the unit attractiveness cost of the competitor's facility k, respectively. The author proves the concavity of the follower' objective function with respect to attractiveness level q. Making use of this property the author transforms the bi-level model into an equivalent single level mixed integer program so that it can be solved by global optimization methods. The transformation is done by substituting the KKT first order conditions into the leader's problem.

The model was further developed in [22] so as to allow the follower to make decisions not only regarding the attractiveness level but also regarding location.

4 Competitive Facility Location with Competition of Customers

The research work dealing with the bi-level formulation of location problems is limited only to the competition among the locators, that is, it is supposed that either both the locator and the allocator are the same or the customer knows the optimality criterion of the locator and agrees passively with it. Customers' preferences as well as externalities such as road congestion, facility congestion and emissions caused by the location decisions are either ignored or "controlled" by incorporating constraints in order to "ensure" the achievement of a predetermined target. However, this approach treats customers as irresolute beings. Thus, if, for example, the customers travel to the facilities to obtain the offered service, then there is no compulsion or incentive for them to attend the designated facility. This means that, once the facilities are open, what the locator wishes the customers to do may not coincide with their own wish and behavior.

The first attempt to study the influence of market competition on location decisions is done by Tobin and Friesz [29]. They analyze the case of a profit maximizing firm which is entering into spatially separated markets and knows that its location decisions will have impact on market prices.

To address the problem they proposed two different models to capture the market competition and its effect on price and production quantities: a spatial price equilibrium (SPE) which determines equilibria in price and production levels for perfectly competitive market and a Cournot Nash oligopolistic model in which a few profit maximizing firms compete in spatially separated markets. They used sensitivity analysis on variational inequalities to relate changes in price to changes in production to obtain optimal locations.

In [16] a bi-level programming model is presented to seek the optimal location for logistics distribution centers. The upper-level model is to determine the optimal location by minimizing the planner's cost and the lower gives an equilibrium demand distribution by minimizing the customer's cost:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}(X_{ij})X_{ij} + \sum_{j=1}^{n} f_j z_j \tag{60}$$

$$st \quad \sum_{J=1}^{n} z_j \geq 1 \tag{61}$$

$$z_j \in \{0,1\} \tag{62}$$

where X_{ij} solves

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} \int_0^{X_{ij}} D^{-1}(w)dw \tag{63}$$

$$st \quad \sum_{j=1}^{n} X_{ij} = w_i, \forall i = 1,\ldots,m, \tag{64}$$

$$\sum_{i=1}^{m} X_{ij} \leq s_j, \forall j = 1,\ldots,n, \tag{65}$$

$$X_{ij} \leq M z_j, \forall i = 1,\ldots,m, j = 1,\ldots,n, \tag{66}$$

$$X_{ij} \geq 0, \forall i = 1,\ldots,m, j = 1,\ldots,n \tag{67}$$

where $C_{ij}(\cdot)$ is the unit generalized cost of meeting the demand of customer i from the distribution center j, and it is usually a nonlinear function; X_{ij} is the demand of the customer i supplied by distribution center j; f_j is the fixed investment associated with building distribution center j; z_j is a $0-1$ variable, if distribution center j is built, then z_j takes the value of 1, and 0 otherwise; $D^1(\cdot)$ is the inverse of demand functions; w_i is the total demand of customer i; s_j is the capacity of distribution center j; M is an arbitrarily large positive constant.

From the point of decision-makers, the first term of objective function (60) represents the total costs of meeting customers' demand. Constraint (61) ensures that at least one distribution center is built, and constraint (62) represents the binary restrictions of the decision variables. The lower-level problem represents the customers' choice behaviors. Constraint (64) ensures that the total demand of each

customer must be met by supply from some distribution centers. Constraint (65) is the capacity constraint, which ensures that all the demands distributed in a distribution center will not exceed its capacity. Constraint (66) prohibits the demand on any proposed distribution center that is not actually constructed. Based on the special form of constraints (66), a simple reaction function is proposed. This reaction function is obtained by transforming (66) into the form

$$X_{ij} = Mz_j - y_{ij}^* \tag{68}$$

where y_{ij}^* is the optimal relaxation variable obtained after solving the second-level problem by any existing algorithm. This reaction function is substituted in the first level of the problem which results to an integer programming problem with variables $\mathbf{z}$ which can be solved by any well-known non-linear programming model

In [17] and [18] the effects of customers' competition for the offered service level on the facility location decisions are examined. Two types of decision-makers are considered, the producer who tries to provide at facilities the best level of service at minimum cost and the customers who make their choices in order to minimize their perceived costs. The customers are involved in a Nash-type game in their effort to ensure the best level of services for themselves. A bi-level programming model is formulated in order to take into consideration the effects of customers' competition. Furthermore an extension is also proposed. It is assumed that there are two producers who constitute a duopoly in the network. The producers compete with one another with respect to the service level they offer in order to attract customers. A bi-level model with two leaders is proposed in order to take into account both the competition between producers and the competition among customers.

It is assumed that the producer tries to provide to the customers the best service level at minimum cost. The evaluation of the offered service is based on the delay faced by the customers at each distribution center i. If x_{ij} is the amount that the customer j buys from the distribution center i, then the performance function $d_i(x_i)$ measures the level of service offered by the distribution center i where $x_i = \sum_{j=1}^{n} x_{ij}$. Suppose that m is the set of potential sites for the location of the distribution centers. We assume that the establishment of a distribution center to the candidate site i implies a fixed location cost f_i. Furthermore, suppose r_j is the demand of customer $j(j = 1\ldots,n)$, p_i is the unit price paid by customers, and q_i is the capacity of the distribution center $i(i = 1\ldots,m)$. Under the assumption that a central coordinator chooses the location of the distribution center in such a manner that the total cost of the system is minimized, the mathematical model can be formulated as follows:

$$(\mathbf{SO}-\mathbf{FL}) \ \min \quad \sum_{i=1}^{m} d_i(x_i)x_i + \sum_{i=1}^{m} p_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{n} t_{ij}x_{ij} + \sum_{i=1}^{m} F_i y_i \tag{69}$$

$$\text{s.t} \quad \sum_{i=1}^{m} x_{ij} = r_j, \ \forall j \tag{70}$$

$$x_i \leq y_i q_i, \ \forall i \tag{71}$$

$$x_i - \sum_{j=1}^{n} x_{ij} = 0, \ \forall i \tag{72}$$

$$y_i \in \{0,1\}, \forall i \tag{73}$$

$$x_{ij} \geq 0, \ \forall i, \ \forall j \tag{74}$$

The objective function of problem (70) minimizes the total cost consisting of the cost of the delay, plus the transportation and purchasing costs plus the cost involved in setting up a distribution center. Constraints (70) ensure that the quantities purchased by the customer j at all distribution centers meet his overall demand. Constraints (71) impose that the total amount of the product available at each distribution center i does not exceed its capacity. In addition, it enables the assignment of the customers' demand only in sited distribution. Relations (72) are the defining constraints of the model, ensuring the maintenance of flow in the network.

In a second model producer takes into account the free will and the competitive preference of the customers and determines the final location of the distribution centers based on the prediction of their behavior as delivered by the outcome of a Nash game. Thus, problem is formulated as bi-level programming model:

$$(\mathbf{BSO}-\mathbf{FL}) \ \min_{[y_i]} \quad \sum_{i=1}^{m} F_i y_i + \sum_{i=1}^{m} d_i(\bar{x}_i)\bar{x}_i + \sum_{i=1}^{m} p_i \bar{x}_i + \sum_{i=1}^{m}\sum_{j=1}^{n} t_{ij}\bar{x}_{ij} \tag{75}$$

$$\text{s.t} \quad y_i \in \{0,1\}, \ \forall i \tag{76}$$

where $[\bar{x}_i]$ and $[\bar{x}_{ij}]$ solve

$$(\mathbf{UO}-\mathbf{TP}) \ \min \quad \sum_{i=1}^{m} \int_0^{x_i} d_i(t)dt + \sum_{i=1}^{m} p_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{n} t_{ij} x_{ij} \tag{77}$$

$$\text{s.t} \quad \sum_{i=1}^{m} x_{ij} = r_j, \ \forall j \tag{78}$$

$$x_i \leq q_i y_i, \ \forall i \tag{79}$$

$$x_i - \sum_{j=1}^{n} x_{ij} = 0, \ \forall i \tag{80}$$

$$x_{ij} \geq 0 \ \forall i,j \tag{81}$$

According to this model, the leader (producer) decides the location of distribution centers solving problem (75)–(76), but he does not control the variables x_i and x_{ij} since they describe the choices of his customers. The values of the variables $[\bar{x}_i]$ and $[\bar{x}_{ij}]$ are derived from model (77)–(81) corresponding to an oracle. In other words, the leader uses (77)–(81) as an oracle to discover trends/reactions of the customers in each potential location and tries to minimize the total cost of the system based on these discoveries.

In a supply chain network where there are more than one producers, none of them have the power to direct customers to distribution centers. Thus, as a result, the offered service level and customer satisfaction are the basic differentiation and discrimination components among economic units of the same sector. In order to take into account both levels of competition we formulate the following bi-level problem with two leaders:

Let us assume that the potential location of distribution centers $i = 1,\ldots,m$ is dispersed between the two producers who in turn are involved in a competition for customer attraction through the provided service level. Let M_1 and M_2 $(m = |M_1| + |M_2|)$ be the nodes of the two producers. Then, under the assumption that both producers "announce their strategies simultaneously," we obtain a Nash game with two players who are dealing (for $K = 1,2$) with the following problems:

The facility location problem of producer 1:

$$(\mathbf{CFL_1}) \ \min \quad \sum_{i\in M_1} F_i y_i + \sum_{i\in M_1} d_i(\bar{x}_i)\bar{x}_i + \sum_{i\in M_1} p_i\bar{x}_i + \sum_{i\in M_1}\sum_{j=1}^{n} t_{ij}\bar{x}_{ij} \tag{82}$$

$$\text{s.t} \quad y_i \in \{0,1\}, \forall i \in M_1 \tag{83}$$

The facility location problem of producer 2:

$$(\mathbf{CFL_2}) \ \min \quad \sum_{i\in M_2} F_i y_i + \sum_{i\in M_2} d_i(\bar{x}_i)\bar{x}_i + \sum_{i\in M_2} p_i\bar{x}_i + \sum_{i\in M_2}\sum_{j=1}^{n} t_{ij}\bar{x}_{ij} \tag{84}$$

$$\text{s.t} \quad y_i \in \{0,1\}, \forall i \in M_2 \tag{85}$$

where $[\bar{x}_i]$ and $[\bar{x}_{ij}]$ solve (77)–(81)

The producers compete with each other with respect to the service level they offer in order to attract customers involved in a Nash game. A Nash equilibrium for this duopolistic game corresponds to a set of location and capacity choices (strategies), which ensure that none of the players are better off by unilaterally changing his strategy.

Let $Y = \{y_i | y_i \in \{0,1\}, \forall i \in M_k\}$ be the feasible sets of the players for $k = 1,2$, $\mathbf{y}_k = [y_i]_{i \in M_k}$ and $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$. We have already mentioned the existence of optimal solutions $\bar{x}_i$ and $\bar{x}_{ij}$ for given capacity $[\bar{q}_i]$. Thus, there is a function from $\mathbb{R}^m$ to $\mathbb{R}^m$, such that for a given $\bar{\mathbf{y}}$ it returns the unique equilibrium point $[\bar{x}_i]$ from (77)–(81) and a corresponding mapping from $\mathbb{R}^m$ to $\mathbb{R}^{m \cdot n}$ such that for a given $\bar{\mathbf{y}}$ it returns an optimal transportation plan $[\bar{x}_{ij}]$ which corresponds to the equilibrium point $[\bar{x}_i]$, thus it holds that $\bar{x}_i = x_i(\bar{\mathbf{y}})$ and $\bar{x}_{ij} = x_{ij}(\bar{\mathbf{y}})$, respectively.

Hence problems $(\mathbf{CFL_k})$ could be formulated as a single level problems:

$$(\mathbf{SCFL_k}) \quad \min_{\mathbf{y}_k \in Y_k} \quad \sum_{i \in M_k} d_i(x_i(\mathbf{y}), y_i) x_i(\mathbf{y}) + \sum_{i \in M_k} p_i x_i(\mathbf{y}) \tag{86}$$

$$+ \sum_{i \in M_k} \sum_{j=1}^{n} t_{ij} x_{ij}(\mathbf{y}) \tag{87}$$

Each problem $(\mathbf{SCFL_k})$ corresponds to player k who is involved in the Nash game.

5 Conclusion and Future Research

The literature concerning the competitive facility location is vast. The main contribution of our study is that it provides a broad review of modeling and optimization approaches of the discrete bi-level version of the problem. The proposed taxonomy can be meaningfully enhanced based on time, evolution, and content of the subject. In addition it could be the basis of a framework for future studies.

References

1. Alekseeva, E., Kocheva, N., Kotcetov, Y., Plyasunov, A.: Heuristic and exact methods for the discrete $(r|p)$- centroid problem. In: Cowling, P., Merz, P. (eds.) Evolutionary Computation in Combinatorial Optimization. Lecture Notes in Computer Science, vol. 6022, pp. 11–22. Springer, Berlin/Heidelberg (2010)
2. Alekseeva, E., Kochetov, Y., Plyasunov, A.: An exact method for the discrete $(r|p)$-centroid problem. J. Glob. Optim. (2013). http://dx.doi.org/10.1007/s10898-013-0130-6
3. Beresnev, V.L.: Upper bounds for objective function of discrete competitive facility location problems. J. Appl. Ind. Math. **3**(4), 3–24 (2009)
4. Beresnev, V.: Branch and bound algorithm for a competitive facility location problem. Comput. Oper. Res. **40**, 2062–2070 (2013)

5. Beresnev, V.L.: On the competitive facility location problem with free choice of supplier. Autom. Remote Control **75**(4), 668–676 (2014)
6. Campos Rodríguez, C.M., Santos Peñate, D.R., Moreno Pérez, J.A.: An exact procedure and lp formulations for the leader-follower location problem. TOP **18**(1), 97–121 (2010)
7. Davydov, I.A, Kochetov, Y.A., Mladenovic, N., Urosevic, D.: Fast metaheuristics for the discrete $(r|p)$-centroid problem. Autom. Remote Control **75**(4), 677–687 (2014)
8. Drezner, T.: Locating a single new facility among existing facilities unequally attractive facilities. J. Reg. Sci. **34**, 237–252 (1994)
9. Drezner, T.: Competitive facility location in plane. In: Drezner, Z. (ed.) Facility Location. A Survey of Applications and Methods, pp. 285–300. Springer, Berlin (1995)
10. Eiselt, H., Laport, G., Thisse, J.F.: Competitive location models: a framework and bibliography. Transp. Sci. **27**, 44–54 (1993)
11. Hakimi, S.L.: On locating new facilities in a competitive environment. Eur. J. Oper. Res **12**, 29–35 (1983)
12. Hakimi, S.L.: Location with spatial interaction. In: Mirchandani, P.B., Francis, R.L. (eds.) Discrete Location Theory, pp. 439–478. Wiley, New York (1990)
13. Hodgson, M.J.: A location-allocation model maximizing consumers welfare. Reg. Stud. **15**, 493–506 (1981)
14. Hotelling, H.: Stability in competition. Econ. J. **39**, 41–57 (1929)
15. Huff, D.: Defining and estimating a trade are. J. Market. **28**, 34–38 (1948)
16. Huijun, S., Ziyou, G., Jianjun, W.: A bi-level programming model and solution algorithm for the location of logistics distribution centers. Appl. Math. Model. **32**, 610–616 (2008)
17. Karakitsiou, A.: Coordination and competition in supply chain: optimization and game theoretic approaches. Ph.D. thesis, Technical University of Crete (2007, in Greek)
18. Karakitsiou, A.: Competitive multilevel capacity allocation. In: Migdalas, A., et al. (eds.) Optimization Theory, Decision Making, and Operations Research Applications. Springer Proceedings in Mathematics & Statistics. Springer, New York (2013)
19. Kress, D., Pesch, E.: Sequential competitive location on networks. Eur. J. Oper. Res **217**(3), 483–499 (2012)
20. Küçkayadin, H., Aras, N., Altinel, I.K.: Competitive facility location problem with attractiveness adjustment of the follower: a bilevel programming model and its solution. Eur. J. Oper. Res. **208**, 206–220 (2011)
21. Küçkayadin, H., Aras, N., Altinel, I.K.: A discrete competitive facility location model with variable attractiveness. J. Oper. Res. Soc. **62**, 1726–1741 (2011)
22. Küçkayadin, H., Aras, N., Altinel, I.K.: A leader-follower game in competitive facility location. Comput. Oper. Res. **39**, 437–448 (2012)
23. Mel'nikov, A.A.: Computational complexity of the discrete competitive facility location. J. Appl. Ind. Math. **8**(4), 557–567 (2014)
24. Mel'nikov, A.A.: Randomized local search for the discrete competitive facility location problem. Autom. Remote Control **75**(4), 700–714 (2014)
25. Nakanishi, M., Cooper, L.G.: Parameters estimation for a multiplicative competitive interaction model: least squares approach. J. Market. Res. **11**, 303–311 (1974)
26. Roboreto, M.C., Pessoa, A.A.: A branch and cut algorithm for the discrete $(r|p)$-centroid problem. Eur. J. Oper. Res. **224**, 101–109 (2013)
27. Serra, D., ReVelle, C.: Competitive location in discrete space. In: Drezner, Z. (ed.) Facility Location. A Survey of Applications and Methods, pp. 367–386. Springer, Berlin (1995)
28. Suárez-Vega, R., Santos-Peñate, D.R., Dorta-González, P.: Competitive multi-facility location on networks: the $(r|x_p)$-medianoid problem. J. Reg. Sci. **44**(3), 569–588 (2004)
29. Tobin, R., Friesz, T.L.: Spatial competition facility location models: definitions, formulations and solution approach. Ann. Oper. Res. **6**, 49–74 (1986)
30. von Stackelberg, H.: The Theory of the Market Economy/by Heinrich von Stackelberg; translated from the German and with an introduction by Alan T. Peacock. William Hodge, London (1952)

On Nash Equilibria in Stochastic Positional Games with Average Payoffs

Dmitrii Lozovanu and Stefan Pickl

Abstract We consider a class of stochastic positional games that extends deterministic positional games with average payoffs. The considered class of games we formulate and study applies the game-theoretical concept to finite state space Markov decision processes with an average cost optimization criterion. Necessary and sufficient conditions for the existence of Nash equilibria in stochastic positional games with average payoffs are proven and some approaches for determining the optimal stationary strategies of the players are analyzed. For antagonistic positional games are proposed. Iterative algorithms for determining the saddle points. Additionally we show that the obtained results can be used for studying the problem of the existence of Nash equilibria in Shapley stochastic games with average payoffs.

Keywords Stochastic positional games • Finite space • Markov processes • Nash equilibrium • Saddle point algorithm • Shapley stochastic games

1 Introduction

We consider a class of stochastic positional games that extends deterministic positional games with average payoffs from [1, 2, 5, 9, 14] and can be used for studying the problem of the existence of Nash equilibria for Shapley stochastic games with average payoffs[18]. The considered class of games we formulate and study applies the concept of positional games to finite state space Markov decision processes with an average cost optimization criterion. We assume that the Markov

D. Lozovanu
Institute of Mathematics and Computer Science, Academy of Sciences, Academy str. 5, Chisinau MD-2028, Moldova
e-mail: lozovanu@math.md

S. Pickl (✉)
Institute for Theoretical Computer Science, Mathematics and Operations Research, Universität der Bundeswehr München, 85577 Neubiberg-München, Germany
e-mail: stefan.pickl@unibw.de

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_9

process is controlled by several actors (players) as follows: The set of states of the system is divided into several disjoint subsets which represent the corresponding position sets of several players. Additionally, the starting position of the game is fixed and the cost of system's transition from one state to another is given for each player separately. Each player has to determine which action should be taken in each state of his position set of Markov process in order to minimize (or maximize) his own average cost per transition. In these games we are seeking a Nash equilibrium.

The main results of the chapter are concerned with the existence of Nash equilibria for the considered class of games and determining the optimal strategies of the players. Necessary and sufficient conditions for the existence of Nash equilibria in stochastic positional games with average payoffs that extend Nash equilibria conditions for deterministic positional games are proven. Based on the constructive proof of these results we propose some approaches for determining the optimal strategies of the players. For the antagonistic positional games we show that a saddle point always exists and an iterative algorithm for determining the optimal stationary strategies of the players is proposed. Additionally we show that the stochastic positional games are tightly connected with Shapley stochastic games and the obtained results can be used for studying the problem of the existence of Nash equilibria for Shapley stochastic games with average payoffs.

2 Problem Formulation and Some Preliminary Results

In this section we formulate the stochastic positional game with average payoffs and describe some preliminary results from [8, 11, 12] that we shall use in the following to prove the basic theorems concerned with the existence of Nash equilibria in this game. We formulate our game model using the framework of a Markov decision process (X,A,p,c) with a finite set of states X, a finite set of actions A, a transition probability function $p : X \times X \times A \to [0,1]$ that satisfies the condition

$$\sum_{y \in X} p^a_{x,y} = 1, \quad \forall x \in X, \quad \forall a \in A$$

and a transition cost function $c : X \times X \to R$ which gives the costs $c_{x,y}$ of state transitions of the dynamical system from an arbitrary state $x \in X$ to an arbitrary state $y \in X$ (see [6, 17]). For the stochastic positional game with m players we assume that m transition cost functions

$$c^i : X \times X \to R, \quad i = 1,2,\ldots,m$$

are given, where $c^i_{x,y}$ expresses the cost of the system's transition from the state $x \in X$ to the state $y \in X$ for the player $i \in \{1,2,\ldots,m\}$. In addition we assume that the set of states X is divided into m disjoint subsets $X_1, X_2, \ldots, X_m$

$$X = X_1 \cup X_2 \cup \cdots \cup X_m \ \ (X_i \cap X_j = \emptyset, \ \ \forall i \neq j),$$

where X_i represents the position set of player $i \in \{1,2,\ldots,m\}$.

The game starts at given position $x_0 = x(0)$ where the player who is the owner of this position starts the game by fixing an action $a \in A(x_0)$. After that the dynamical system passes randomly to the next position $x(1)$ according to distribution transition probabilities that correspond to a selected action a in x_0. Then the player who is the owner of new position $x(1)$ fixes an action $a \in A(x(1))$ and the dynamical system passes randomly to the next position $x(2)$ according to distribution transition probabilities that correspond to the action a in $x(1)$ and so on indefinitely. Each player in this game selects actions in his position set in order to minimize his own average cost per transition. In the considered game we are seeking for a Nash equilibrium.

More precisely we can formulate the considered stochastic positional game in terms of stationary strategies. In the following we can see that if a Nash equilibrium exists in this game then it is reached in the set of stationary strategies. The stationary strategies of the players we define as m maps:

$$s^i : x \to a \in A^i(x) \ \text{ for } \ x \in X_i, \ i = 1,2,\ldots,m,$$

where $A^i(x)$ is the set of actions of the player i in the state $x \in X_i$. Without loss of generality we may consider $|A^i(x)| = |A^i| = |A|, \ \forall x \in X_i, \ i = 1,2,\ldots,m$. In order to simplify the notation we denote the set of possible actions in a state $x \in X$ for an arbitrary player by $A(x)$. A stationary strategy s^i, $i \in \{1,2,\ldots,m\}$ in the state $x \in X_i$ means that at every discrete moment of time $t = 0,1,2,\ldots$ the player i uses the action $a = s^i(x)$. Players fix their strategy independently and do not inform each other which strategies they use in the decision process.

If the players $1,2,\ldots,m$ fix their stationary strategies $s^1, s^2, \ldots, s^m$, respectively, then we obtain a situation $s = (s^1, s^2, \ldots, s^m)$. This situation corresponds to a simple Markov process determined by the probability distributions $p_{x,y}^{s^i(x)}$ in the states $x \in X_i$ for $i = 1,2,\ldots,m$. We denote by $P^s = (p_{x,y}^s)$ the matrix of probability transitions of this Markov process. If the starting state x_0 is given, then for the Markov process with the matrix of probability transitions P^s we can determine the average cost per transition $\omega_{x_0}^i(s^1, s^2, \ldots, s^m)$ with respect to each player $i \in \{1,2,\ldots,m\}$ taking into account the corresponding matrix of transition costs $C^i = (c_{x,y}^i)$. So, on the set of situations we can define the payoff functions of the players as follows:

$$F_{x_0}^i(s^1, s^2, \ldots, s^m) = \omega_{x_0}^i(s^1, s^2, \ldots, s^m), \quad i = 1,2,\ldots,m.$$

In such a way we obtain a discrete noncooperative game in normal form which is determined by a finite set of strategies $S^1, S^2, \ldots, S^m$ of m players and the payoff functions defined above. In this game we are seeking a *Nash*

equilibrium [15], i.e., we consider the problem of determining the stationary strategies $s^{1^*}, s^{2^*}, \ldots, s^{i-1^*}, s^{i^*}, s^{i+1^*}, \ldots, s^{m*}$ such that

$$F^i_{x_0}(s^{1^*}, s^{2^*}, \ldots, s^{i-1^*}, s^{i^*}, s^{i+1^*}, \ldots, s^{m*}) \leq F^i_{x_0}(s^{1^*}, s^{2^*}, \ldots, s^{i-1^*}, s^i, s^{i+1^*}, \ldots, s^{m*}), \ \forall s^i \in S^i, \ i = 1, 2, \ldots, m.$$

The game defined above is determined uniquely by the set of states X, the position sets $X_1, X_2, \ldots, X_m$, the set of actions A, the cost functions, $c^i : X \times X \to R$, $i = 1, 2, \ldots, m$, the probability function $p : X \times X \times A \to [0, 1]$ and the starting position x_0. Therefore, we denote this game $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, x_0)$. In the case $m = 2$ and $c^2 = -c^1$ we obtain an antagonistic stochastic positional game. If $p^a_{x,y} = 0 \vee 1$, $\forall x, y \in X$, $\forall a \in A$ the stochastic positional game $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, x_0)$ is transformed into the cyclic game [5, 9, 10]. Some results concerned with the existence of Nash equilibria for stochastic positional games with average payoffs have been derived in [8, 11, 12]. In particular the following theorem has been proven [8].

Theorem 1. *If for an arbitrary situation $s = (s^1, s^2, \ldots, s^m)$ of the stochastic positional game with average payoffs the matrix of probability transitions $P^s = (p^s_{x,y})$ induces an ergodic Markov chain then for the game there exists a Nash equilibrium.*

Based on a constructive proof of this theorem in [8] an algorithm is proposed for determining the optimal stationary strategies of the players if the conditions of theorem hold. If the matrix P^s for some situations do not correspond to an ergodic Markov chain then for the stochastic positional game with average payoffs a Nash equilibrium may not exist. This follows from the constructive proof of this theorem. An example of a deterministic positional game with average payoffs for which Nash equilibrium does not exist has been constructed in [5]. However, in the following we can see that for an arbitrary antagonistic stochastic positional games saddle points always exist.

Note that in [8, 12] studied also stochastic positional games with discounted payoffs have been and it is shown that for such games a Nash equilibrium always exists.

3 Nash Equilibria Conditions for Stochastic Positional Games with Average Payoffs

The aim of this section is to formulate Nash equilibria conditions for stochastic positional games in terms of bias equations for Markov decision processes. We can see that Nash equilibria conditions in such terms may be useful for determining the optimal strategies of the players.

Theorem 2. *Let* $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, \bar{x})$ *be a stochastic positional game with a given starting position* $\bar{x} \in X$ *and average payoff functions*

$$F^1_{\bar{x}}(s^1, s^2, \ldots, s^m),\ F^2_{\bar{x}}(s^1, s^2, \ldots, s^m),\ \ldots,\ F^m_{\bar{x}}(s^1, s^2, \ldots, s^m)$$

of the players $1, 2, \ldots, m$*, respectively. Assume that for an arbitrary situation* $s = (s^1, s^2, \ldots, s^m)$ *of the game the transition probability matrix* $P^s = (p^s_{x,y})$ *corresponds to an ergodic Markov chain. Then there exist the functions*

$$\varepsilon^i : X \to R, \quad i = 1, 2, \ldots, m$$

and the values $\omega^1, \omega^2, \ldots, \omega^m$ *that satisfy the following conditions:*

1) $\mu^i_{x,a} + \sum\limits_{y \in X} p^a_{x,y}\varepsilon^i_y - \varepsilon^i_x - \omega^i \geq 0,\ \forall x \in X_i, \quad \forall a \in A(x),\ i = 1, 2, \ldots, m,$
   where $\mu^i_{x,a} = \sum\limits_{y \in X} p^a_{x,y} c^i_{x,y}$;
2) $\min\limits_{a \in A(x)} \{\mu^i_{x,a} + \sum\limits_{y \in X} p^a_{x,y}\varepsilon^i_y - \varepsilon^i_x - \omega^i\} = 0, \quad \forall x \in X_i,\ i = 1, 2, \ldots, m;$
3) *on each position set* X_i, $i \in \{1, 2, \ldots, m\}$ *there exists a map* $s^{i*} : X_i \to A$ *such that*

$$s^{i*}(x) = a^* \in Arg \min_{a \in A(x)} \left\{\mu^i_{x,a} + \sum_{y \in X} p^a_{x,y}\varepsilon^i_y - \varepsilon^i_x - \omega^i\right\}$$

and

$$\mu^j_{x,a^*} + \sum_{y \in X} p^{a^*}_{x,y}\varepsilon^j_y - \varepsilon^j_x - \omega^j = 0,\ \ \forall x \in X_i,\ \ j = 1, 2, \ldots, m.$$

The set of maps $s^{1*}, s^{2*}, \ldots, s^{m*}$ *determines a Nash equilibrium situation* $s^* = (s^{1*}, s^{2*}, \ldots, s^{m*})$ *for the stochastic positional game* $\left(X,\ A,\ \{X_i\}_{i=\overline{1,m}},\ \{c^i\}_{i=\overline{1,m}},\ p,\ \bar{x}\right)$ *and*

$$F^i_{\bar{x}}(s^{1*}, s^{2*}, \ldots, s^{m*}) = \omega^i,\ \ \forall \bar{x} \in X,\ i = 1, 2, \ldots, m.$$

Moreover, the situation $s^* = (s^{1*}, s^{2*}, \ldots, s^{m*})$ *is a Nash equilibrium for an arbitrary starting position* $\bar{x} \in X$.

Proof. Let a stochastic positional game with average payoffs be given and assume that for an arbitrary situation s of the game the transition probability matrix $P^s = (p^s_{x,y})$ corresponds to an ergodic Markov chain. Then according to Theorem 1 for this game there exists a Nash equilibrium $s^* = (s^{1*}, s^{2*}, \ldots, s^{m*})$ and we can set

$$\omega^i = F^i_{\bar{x}}(s^{1*}, s^{2*}, \ldots, s^{m*}),\ \ \forall \bar{x} \in X,\ i = 1, 2, \ldots, m.$$

Let us fix the strategies $s^{1^*}, s^{2^*}, \ldots, s^{i-1^*}, s^{i+1^*}, \ldots, s^{m^*}$ of the players $1, 2, \ldots, i-1, i+1, \ldots, m$ and consider the problem of determining the minimal average cost per transition with respect to player i. Obviously, if we solve this decision problem then we obtain the strategy s^{i^*}. We can determine the optimal strategy of this decision problem with average cost optimization criterion using the bias equations with respect to player i. This means that there exist the functions $\varepsilon^i : X \to R$ and the values ω^i, $i = 1, 2, \ldots, m$ that satisfy the conditions:

1) $\mu^i_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i \geq 0, \quad \forall x \in X_i, \ \forall a \in A(x);$

2) $\min\limits_{a \in A(x)} \left\{ \mu^i_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i \right\} = 0, \quad \forall x \in X_i.$

Moreover, for fixed strategies $s^{1^*}, s^{2^*}, \ldots, s^{i-1^*}, s^{i+1^*}, \ldots, s^{m^*}$ of the corresponding players $1, 2, \ldots, i-1, i+1, \ldots, m$ we can select the strategy s^{i^*} of player i where

$$s^{i^*}(x) \in Arg \min_{a \in A(x)} \left\{ \mu^i_{x,a} + \sum_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i \right\}$$

and $\omega^i = F^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m^*})$, $\forall x \in X$, $i = 1, 2, \ldots, m$. This means that conditions 1)–3) of the theorem holds. □

Corollary 1. *If for a stochastic positional game* $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, x)$ *with average payoffs there exists a Nash equilibrium* $s^* = (s^{1^*}, s^{2^*}, \ldots, s^{m^*})$ *which is a Nash equilibrium for an arbitrary starting position of the game* $x \in X$ *and for arbitrary two different starting positions* $x, y \in X$ *it holds* $F^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m^*}) = F^i_y(s^{1^*}, s^{2^*}, \ldots, s^{m^*})$ *then exist the functions*

$$\varepsilon^i : X \to R, \quad i = 1, 2, \ldots, m$$

and the values $\omega^1, \omega^2, \ldots, \omega^m$ *that satisfy the conditions 1)–3) from Theorem 2. So,* $\omega^i = F^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m^*})$, $\forall x \in X$, $i = 1, 2, \ldots, m$ *and an arbitrary Nash equilibrium can be found by fixing*

$$s^{i^*}(x) = a^* \in Arg \min_{a \in A(x)} \left\{ \mu^i_{x,a} + \sum_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i \right\}.$$

Using the elementary properties of non-ergodic Markov decision processes with average cost optimization criterion the following lemma can be gained.

Lemma 1. *Let* $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, x)$ *be an average stochastic positional game for which there exists a Nash equilibrium* $s^* = (s^{1^*}, s^{2^*}, \ldots, s^{m^*})$, *which is a Nash equilibrium for an arbitrary starting position of the game with* $\omega^i_x = F^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m^*})$. *Then* $s^* = (s^{1^*}, s^{2^*}, \ldots, s^{m^*})$ *is a Nash equilibrium for the average stochastic positional game* $(X, A, \{X_i\}_{i=\overline{1,m}}, \{\overline{c}^i\}_{i=\overline{1,m}}, p, x)$, *where*

$$\overline{c}^i_{x,y} = c^i_{x,y} - \omega^i_x, \ \forall x, y \in X, \ \ i = 1, 2, \ldots, m$$

and

$$\overline{F}^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m*}) = 0, \ \forall x \in X, \ \ i = 1, 2, \ldots, m.$$

Now using Corollary 1 and Lemma 1 we can prove the following results.

Theorem 3. *Let* $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, x)$ *be an average stochastic positional game. Then in this game there exists a Nash equilibrium for an arbitrary starting position* $x \in X$ *if and only if there exist the functions*

$$\varepsilon^i : X \to R, \ i = 1, 2, \ldots, m$$

and the values $\omega^1_x, \omega^2_x, \ldots, \omega^m_x$ *for* $x \in X$ *that satisfy the following conditions:*

1) $\mu^i_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i_x \geq 0, \quad \forall x \in X_i, \ \forall a \in A(x), \ i = 1, 2, \ldots, m,$
   where $\mu^i_{x,a} = \sum\limits_{y \in X} p^a_{x,y} c^i_{x,y}$;
2) $\min\limits_{a \in A(x)} \{\mu^i_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i_x\} = 0, \quad \forall x \in X_i, \ i = 1, 2, \ldots, m;$
3) *on each position set* X_i, $i \in \{1, 2, \ldots, m\}$ *there exists a map* $s^{i^*} : X_i \to A$ *such that*

$$s^{i^*}(x) = a^* \in Arg \min_{a \in A(x)} \left\{ \mu^i_{x,a} + \sum_{y \in X} p^a_{x,y} \varepsilon^i_y - \varepsilon^i_x - \omega^i \right\}$$

and

$$\mu^j_{x,a^*} + \sum_{y \in X} p^{a^*}_{x,y} \varepsilon^j_y - \varepsilon^j_x - \omega^j = 0, \quad \forall x \in X_i, \ \ j = 1, 2, \ldots, m.$$

If such conditions hold then the set of maps $s^{1^*}, s^{2^*}, \ldots, s^{m*}$ *determines a Nash equilibrium of the game for an arbitrary starting position* $x \in X$ *and*

$$F^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m*}) = \omega^i_x, \ i = 1, 2, \ldots, m.$$

Proof. The sufficiency condition of the theorem is evident. Let us prove the necessity one. Assume that for the considered average stochastic positional game there exists a Nash equilibrium $s^* = (s^{1^*}, s^{2^*}, \ldots, s^{m*})$ which is a Nash equilibrium for an arbitrary starting position of the game. Denote

$$\sigma^i_x = \hat{F}^i_x(s^{1^*}, s^{2^*}, \ldots, s^{m*}), \ \ \forall x \in X, \ \ i = 1, 2, \ldots, m$$

and consider the following auxiliary game $(X, A, \{X_i\}_{i=\overline{1,m}}, \{\overline{c}^i\}_{i=\overline{1,m}}, p, x)$, where

$$\overline{c}^i_{x,y} = c^i_{x,y} - \omega^i_x, \ \forall x, y \in X, \ \ i = 1, 2, \ldots, m.$$

Then according to Lemma 1 the auxiliary game has the same Nash equilibrium $s^* = (s^{1^*}, s^{2^*}, \ldots, s^{m*})$ as the initial one. Moreover, this equilibrium is a Nash equilibrium for an arbitrary starting position of the game and

$$\overline{F}_x^i(s^{1^*}, s^{2^*}, \ldots, s^{m*}) = 0, \ \forall x \in X, \ \ i = 1, 2, \ldots, m.$$

Therefore, according to Corollary 1, for the auxiliary game there exist the functions

$$\varepsilon^i : X \to R, \quad i = 1, 2, \ldots, m$$

and the values $\overline{\omega}^1, \overline{\omega}^2, \ldots, \overline{\omega}^m$ ($\overline{\omega}^i = 0, \ i = 1, 2, \ldots, m$) that satisfy the conditions of Theorem 2, i.e.,

1) $\overline{\mu}_{x,a}^i + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y^i - \varepsilon_x^i - \overline{\omega}_x^i \geq 0, \quad \forall x \in X_i, \ \forall a \in A(x), \ i = 1, 2, \ldots, m,$
   where $\overline{\mu}_{x,a}^i = \sum\limits_{y \in X} p_{x,y}^a \overline{c}_{x,y}^i;$
2) $\min\limits_{a \in A(x)} \{\overline{\mu}_{x,a}^i + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y^i - \varepsilon_x^i - \overline{\omega}_x^i\} = 0, \quad \forall x \in X_i, \ i = 1, 2, \ldots, m;$
3) on each position set X_i, $i \in \{1, 2, \ldots, m\}$ there exists a map $s^{i^*} : X_i \to A$ such that

$$s^{i^*}(x) = a^* \in Arg \min_{a \in A(x)} \left\{ \overline{\mu}_{x,a}^i + \sum_{y \in X} p_{x,y}^a \varepsilon_y^i - \varepsilon_x^i - \overline{\omega}^i \right\}$$

and

$$\overline{\mu}_{x,a^*}^j + \sum_{y \in X} p_{x,y}^{a^*} \varepsilon_y^j - \varepsilon_x^j - \overline{\omega}^j = 0, \quad \forall x \in X_i, \ \ j = 1, 2, \ldots, m.$$

Taking into account that $\overline{\omega}_x^i = 0$, and $\overline{\mu}_{x,a}^i = \mu_{x,a}^i - \omega_x^i$ (because $\overline{c}_{x,y}^i = c_{x,y} - \omega_x^i$) we obtain conditions 1)–3) of the theorem. □

4 Saddle Point Conditions for Antagonistic Stochastic Positional Games and an Algorithm for Determining the Optimal Strategies

The antagonistic stochastic positional game with the average payoff corresponds to the case of the game from Sect. 2 in the case $m = 2$ when $c = c^1 = -c^2$. So, we have a game $(X, A, X_1, X_2, c, p, \bar{x})$ where the stationary strategies s^1 and s^2 of the players are defined as two maps

$$s^1 : x \to a \in A^1(x) \ \text{ for } \ x \in X_1; \ \ s^2 : x \to a \in A^1(x) \ \text{ for } \ x \in X_2$$

and the payoff function $F_x(s^1,s^2) = F_x^1(s^1,s^2) = -F_x^2(s^1,s^2)$ of the players is determined by the values of average costs ω_x^s in the Markov processes with the corresponding probability matrices P^s induced by the situations $s = (s^1,s^2) \in S$. We show that for this game saddle points s^{1^*}, s^{2^*} always exist, i.e. for a given starting position $\bar{x} \in X$ it holds

$$F_{\bar{x}}(s^{1^*}, s^{2^*}) = \min_{s^1 \in S^1} \max_{s^2 \in S^2} F_{\bar{x}}(s^1,s^2) = \max_{s^2 \in S^2} \min_{s^1 \in S^1} F_{\bar{x}}(s^1,s^2).$$

Theorem 4. *Let $(X, A, X_1, X_2, c, p, \bar{x})$ be an arbitrary antagonistic stochastic positional game with average payoff function $F_{\bar{x}}(s_1,s_2)$. Then the system of equations*

$$\begin{cases} \varepsilon_x + \omega_x = \max\limits_{a \in A(x)} \left\{ \mu_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon_y \right\}, & \forall x \in X_1; \\ \\ \varepsilon_x + \omega_x = \min\limits_{a \in A(x)} \left\{ \mu_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon_y \right\}, & \forall x \in X_2; \end{cases}$$

has a solution under the set of solutions of the system of equations

$$\begin{cases} \omega_x = \max\limits_{a \in A(x)} \left\{ \sum\limits_{y \in X} p^a_{x,y} \omega_y \right\}, & \forall x \in X_1; \\ \\ \omega_x = \min\limits_{a \in A(x)} \left\{ \sum\limits_{y \in X} p^a_{x,y} \omega_y \right\}, & \forall x \in X_2, \end{cases}$$

i.e. the last system of equations has such a solution ω_x^*, $x \in X$ for which there exists a solution ε_x^*, $x \in X$ of the following system of equations

$$\begin{cases} \varepsilon_x + \omega_x^* = \max\limits_{a \in A(x)} \left\{ \mu_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon_y \right\}, & \forall x \in X_1; \\ \\ \varepsilon_x + \omega_x^* = \min\limits_{a \in A(x)} \left\{ \mu_{x,a} + \sum\limits_{y \in X} p^a_{x,y} \varepsilon_y \right\}, & \forall x \in X_2. \end{cases}$$

The optimal stationary strategies of players

$$s_1{}^* : x \to a^1 \in A(x) \text{ for } x \in X_1;$$
$$s_2{}^* : x \to a^2 \in A(x) \text{ for } x \in X_2$$

in the antagonistic stochastic positional game can be found by fixing arbitrary maps $s_1{}^*(x) \in A(x)$ for $x \in X_1$ and $s_2{}^*(x) \in A(x)$ for $x \in X_2$ such that

$$s_1{}^*(x) \in \left(Arg \max_{a\in A(x)} \left\{ \sum_{y\in X} p^a_{x,y}\omega^*_y \right\}\right) \bigcap \left(Arg \max_{a\in A(x)} \left\{ \mu_{x,a} + \sum_{y\in X} p^a_{x,y}\varepsilon^*_y \right\}\right)$$
$$\forall x \in X_1$$

and

$$s_2{}^*(x) \in \left(Arg \min_{a\in A(x)} \left\{ \sum_{y\in X} p^a_{x,y}\omega^*_y \right\}\right) \bigcap \left(Arg \min_{a\in A(x)} \left\{ \mu_{x,a} + \sum_{y\in X} p^a_{x,y}\varepsilon^*_y \right\}\right).$$
$$\forall x \in X_2$$

For the strategies s^{1^*}, s^{2^*} *the corresponding values of the payoff function* $F_{\bar{x}}(s^{1^*}, s^{2^*})$ *coincides with the values* $\omega^*_{\bar{x}}$ *for* $\bar{x} \in X$ *and*

$$F_x(s^{1^*}, s^{2^*}) = \min_{s^1\in S^1} \max_{s^2\in S^2} F_x(s^1, s^2) = \max_{s^2\in S^2} \min_{s^1\in S^1} F_x(s^1, s^2) \ \ \forall x \in X.$$

.

Proof. Let $\bar{x} \in X$ be an arbitrary state and consider the stationary strategies $\bar{s}^1 \in S^1$, $\bar{s}^2 \in S^2$ for which

$$F_{\bar{x}}(\bar{s}^1, \bar{s}^2) = \min_{s^2\in S^2} \max_{s^1\in S^1} F_{\bar{x}}(s^1, s^2).$$

We show that

$$F_{\bar{x}}(\bar{s}^1, \bar{s}^2) = \max_{s^1\in S^1} \min_{s^2\in S^2} F_{\bar{x}}(s^1, s^2),$$

i.e., we show that $\bar{s}^1 = s^{1^*}$, $\bar{s}^2 = s^{2^*}$.

According to the properties of the bias equations for the situation $\bar{s} = (\bar{s}^1, \bar{s}^2)$ the system of linear equations

$$\begin{cases} \varepsilon_x + \omega_x = \mu_{x,a} + \sum_{y\in X} p^a_{x,y}\varepsilon_y, & \forall x \in X_1,\ a = \bar{s}^1(x); \\ \varepsilon_x + \omega_x = \mu_{x,a} + \sum_{y\in X} p^a_{x,y}\varepsilon_y, & \forall x \in X_2,\ a = \bar{s}^2(x); \\ \omega_x = \sum_{y\in X} p^a_{x,y}\omega_y, & \forall x \in X_1,\ a = \bar{s}^1(x); \\ \omega_x = \sum_{y\in X} p^a_{x,y}\omega_y, & \forall x \in X_2,\ a = \bar{s}^2(x) \end{cases} \tag{1}$$

has the solution $\varepsilon^*_x, \omega^*_x\,(x \in X)$ which for a fixed strategy $\bar{s}^2 \in S^2$ satisfies the condition:

$$\begin{cases} \varepsilon_x^* + \omega_x^* \geq \mu_{x,a} + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y^*, & \forall x \in X_1,\ a \in A(x); \\ \varepsilon_x^* + \omega_x^* = \mu_{x,a} + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y^*, & \forall x \in X_2,\ a = \overline{s}^2(x); \\ \omega_x^* \geq \sum\limits_{y \in X} p_{x,y}^a \omega_y^*, & \forall x \in X_1,\ a \in A(x); \\ \omega_x^* = \sum\limits_{y \in X} p_{x,y}^a \omega_y^*, & \forall x \in X_2,\ a = \overline{s}^2(x) \end{cases}$$

and $F_x(\overline{s}^1, \overline{s}^2) = \omega_x^*,\ \forall x \in X$.

Taking into account that $F_x(\overline{s}^1, \overline{s}^2) = \min\limits_{s^2 \in S^2} F_x(\overline{s}^1, s^2)$ then for a fixed strategy $\overline{s}^1 \in S^1$ the solution ε_x^*, ω_x^* $(x \in X)$ satisfies the condition

$$\begin{cases} \varepsilon_x^* + \omega_x^* = \mu_{x,a} + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y^*, & \forall x \in X_1,\ a = \overline{s}^1(x); \\ \varepsilon_x^* + \omega_x^* \leq \mu_{x,a} + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y^*, & \forall x \in X_2,\ a \in A(x); \\ \omega_x^* = \sum\limits_{y \in X} p_{x,y}^a \omega_y^*, & \forall x \in X_1,\ a = \overline{s}^1(x); \\ \omega_x^* \leq \sum\limits_{y \in X} p_{x,y}^a \omega_y^*, & \forall x \in X_2,\ a \in A(x) \end{cases}$$

So, the following system

$$\begin{cases} \varepsilon_x + \omega_x \geq \mu_{x,a} + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y, & \forall x \in X_1,\ a \in A(x); \\ \varepsilon_x + \omega_x \leq \mu_{x,a} + \sum\limits_{y \in X} p_{x,y}^a \varepsilon_y, & \forall x \in X_2,\ a \in A(x); \\ \omega_x \geq \sum\limits_{y \in X} p_{x,y}^a \omega_y, & \forall x \in X_1,\ a \in A(x); \\ \omega_x \leq \sum\limits_{y \in X} p_{x,y}^a \omega_y, & \forall x \in X_2,\ a \in A(x) \end{cases}$$

has a solution, which satisfies condition (1).

This means that ${s^1}^* = \overline{s}^1, {s^2}^* = \overline{s}^2$ and

$$\max_{s^1 \in S^1} \min_{s^2 \in S^2} F_{\overline{x}}(s^1, s^2) = \min_{s^2 \in S^2} \max_{s^1 \in S^1} F_{\overline{x}}(s^1, s^2),\ \forall \overline{x} \in X,$$

i.e., the theorem holds. □

Based on Theorem 4 we can propose the following algorithm for determining the optimal stationary strategies of the players in the antagonistic stochastic positional game with average payoff function.

4.1 An Algorithm

Preliminary step (Step 0): Fix the arbitrary stationary strategies

$$s_1^0 : x \to a \in A(x) \ \text{ for } \ x \in X_1$$
$$s_2^0 : x \to a \in A(x) \ \text{ for } \ x \in X_2$$

that determine the situation $s^0 = (s_1^0, s_2^0)$.

General step (Step k, $k \geq 1$): Determine the probability matrix $P^{s^{k-1}}$ that corresponds to the situation $s = (s_1^{k-1}, s_2^{k-1})$ and find $\omega^{s^{k-1}}$ and $\varepsilon^{s^{k-1}}$ which satisfy the conditions

$$\begin{cases} (P^{s^{k-1}} - I)\omega^{s^{k-1}} = 0; \\ \mu^{s^{k-1}} + (P^{s^{k-1}} - I)\varepsilon^{s^{k-1}} - \omega^{s^{k-1}} = 0. \end{cases}$$

Then find a situation $s^k = (s_1^k, s_2^k)$ such that

$$s_1^k(x) \in Arg \max_{a \in A(x)} \left\{ \sum_{y \in X} p_{x,y}^a \omega_y^{s_1^{k-1}} \right\}, \qquad \forall x \in X_1;$$
$$s_2^k(x) \in Arg \min_{a \in A(x)} \left\{ \sum_{y \in X} p_{x,y}^a \omega_y^{s_2^{k-1}} \right\}, \qquad \forall x \in X_2$$

and set $s^k = s^{k-1}$ if

$$s_1^{k-1}(x) \in Arg \max_{a \in A(x)} \left\{ \sum_{y \in X} p_{x,y}^a \omega_y^{s_1^{k-1}} \right\}, \qquad \forall x \in X_1;$$
$$s_2^{k-1}(x) \in Arg \min_{a \in A(x)} \left\{ \sum_{y \in X} p_{x,y}^a \omega_y^{s_2^{k-1}} \right\}, \qquad \forall x \in X_2$$

After that check if $s^k = s^{k-1}$. If $s^k = s^{k-1}$ then go to next step $k+1$; otherwise choose a situation $s^k = (s_1^k, s_2^k)$ such that

$$s_1^k(x) \in Arg \max_{a \in A(x)} \left\{ \mu_{x,a} + \sum_{y \in X} p_{x,y}^a \varepsilon_y^{s_1^{k-1}(x)} \right\} \qquad \forall x \in X_1;$$
$$s_2^k(x) \in Arg \min_{a \in A(x)} \left\{ \mu_{x,a} + \sum_{y \in X} p_{x,y}^a \varepsilon_y^{s_2^{k-1}(x)} \right\} \qquad \forall x \in X_2$$

and set $s^k = s^{k-1}$ if

$$s_1^{k-1}(x) \in Arg \max_{a \in A(x)} \left\{ \mu_{x,a} + \sum_{y \in X} p_{x,y}^a \varepsilon_y^{s_1^{k-1}(x)} \right\} \qquad \forall x \in X_1;$$

$$s_2^{k-1}(x) \in Arg \min_{a \in A(x)} \left\{ \mu_{x,a} + \sum_{y \in X} p_{x,y}^a \varepsilon_y^{s_2^{k-1}(x)} \right\} \qquad \forall x \in X_2$$

After that check if $s^k = s^{k-1}$. If $s^k = s^{k-1}$ then STOP and set $s^* = s^{k-1}$; otherwise go to next step $k+1$.

Remark. If $p_{x,y} \in \{0,1\} \forall x, y \in X$ then the algorithm is transformed in the algorithm for determining the optimal stationary strategies of the players in the deterministic parity games.

The convergence of the algorithm described above can be grounded in a similar way as the convergence of the iterative algorithm for determining the optimal solution of the Markov decision problem with average cost optimization criteria (see [6, 17]).

5 Application of Stochastic Positional Games for Studying Shapley Stochastic Games

The aim of this section is to show on the relationship between stochastic positional games and Shapley stochastic games [18]. Based on this relationship and the obtained results we show that in general for Shapley stochastic games with average payoffs a Nash equilibrium may not exist.

A stochastic game in the sense of Shapley [18] is a dynamic game with probabilistic transitions played by several players in a sequence of stages, where the beginning of each stage corresponds to a state of the dynamical system. The game starts at a given state from the set of states of the system. At each stage players select actions from their feasible sets of actions and each player receives a stage payoff that depends on the current state and the chosen actions. The game then moves to a new random state the distribution of which depends on the previous state and the actions chosen by the players. The procedure is repeated at a new state and the play continues for a finite or infinite number of stages. The total payoff of a player is either the limit inferior of the average of the stage payoffs or the discounted sum of the stage payoffs.

So, an average Shapley stochastic game with m players consists of the following elements:

1. A state space X (which we assume to be finite);
2. A finite set $A^i(x)$ of actions with respect to each player $i \in \{1,2,\ldots,m\}$ for an arbitrary state $x \in X$;
3. A stage payoff $f^i(x,a)$ with respect to each player $i \in \{1,2,\ldots,m\}$ for each state $x \in X$ and for an arbitrary action vector $a \in \prod_i A^i(x)$;

4. A transition probability function $p : X \times \prod_{x\in X} \prod_i A^i(x) \times X \to [0,1]$ that gives the probability transitions $p^a_{x,y}$ from an arbitrary $x \in X$ to an arbitrary $y \in Y$ for a fixed action vector $a \in \prod_i A^i(x)$, where

$$\sum_{y\in X} p^a_{x,y} = 1, \quad \forall x \in X,\ a \in \prod_i A^i(x);$$

5. A starting state $x_0 \in X$.

The stochastic game starts in state x_0. At stage t players observe state x_t and simultaneously choose actions $a^i_t \in A^i(x_t), \quad i = 1,2,\ldots,m$. Then nature selects state x_{t+1} according to probability transitions $p^{a_t}_{x_t,y}$ for a fixed action vector $a_t = (a^1_t, a^2_t, \ldots, a^m_t)$. A play of the stochastic game $x_0, a_0, x_1, a_1, \ldots, x_t, a_t, \ldots$ defines a stream of payoffs $f^i_0, f^i_1, f^i_2, \ldots$, where $f^i_t = f^i(x_t, a_t),\ t = 0,1,2,\ldots$. The t-stage average stochastic game is the game where the payoff of player $i \in \{1,2,\ldots,m\}$ is

$$F^i_t = \frac{1}{t} \sum_{\tau=1}^{t-1} f^i_\tau.$$

The infinite average stochastic game is the game where the payoff of player $i \in \{1,2,\ldots,m\}$ is

$$\overline{F}^i = \lim_{t\to\infty} F^i_t.$$

In a similar way is a Shapley stochastic game with expected discounted payoffs of the players is defined. In such a game along with elements described above also a discount factor λ $(0 < \lambda < 1)$ is given and the total payoff of a player represents the expected discounted sum of the stage payoffs.

By comparing Shapley stochastic games with stochastic positional games we can observe the following. The probability transitions from a state to other states as well as the stage payoffs of the players in a Shapley stochastic game depend on the actions chosen by all players, while the probability transitions from a state to other states as well the stage payoffs (the immediate costs of the players) in a stochastic positional game depend only on the action of the player that controls the state in his position set. This means that a stochastic positional game can be regarded as a special case of the Shapley stochastic game. Nevertheless we can see that stochastic positional games can be used for studying some classes of Shapley stochastic games.

The main results concerned with determining Nash equilibria in Shapley stochastic games have been obtained in [3, 4, 7, 13, 16]. The existence of Nash equilibria for such games is proven in the case of stochastic games with a finite set of stages and in the case of the games with infinite stages when the total payoff of each player is the discounted sum of stage payoffs. If the total payoff of a player represents the limit inferior of the average of the stage payoffs then the existence of a Nash equilibrium in Shapley stochastic games is an open question. Based on results mentioned in previous sections we can show that in the case of the average non-antagonistic stochastic games a Nash equilibrium may not

exist. In order to prove this we can use the average stochastic positional game $(X, A, \{X_i\}_{i=\overline{1,m}}, \{c^i\}_{i=\overline{1,m}}, p, x_0)$ from Sect. 2. It is easy to observe that this game can be regarded as a Shapley stochastic game with average payoff functions of the players, where for a fixed situation $s = (s^1, s^2, \ldots, s^m)$ the probability transition $p^s_{x,y}$ from a state $x = x(t) \in X_i$ to a state $y = x(t+1) \in X$ depends only on strategy s^i of player i and the corresponding stage payoff in the state x of player $i \in \{1, 2, \ldots, m\}$ is equal to $\sum_{y \in X} p^s_{x,y} c^i_{x,y}$. Taking into account that the cyclic game represents a particular case of the average stochastic positional game and for the cyclic game a Nash equilibrium may not exist (see example from [5]) we obtain that for the average non-antagonistic Shapley stochastic game a Nash equilibrium may not exist.

6 Conclusion

Stochastic positional games with average payoffs represent a special class of Shapley stochastic games. The obtained results in this chapter show that for the considered positional games a Nash equilibrium may not exist. This involves that for Shapley stochastic games with average payoffs in general case a Nash equilibrium also may not exist. However necessary and sufficient conditions for the existence of Nash equilibria in stochastic positional games with average payoff are derived. For antagonistic positional games it is proven that saddle points always exist and an iterative algorithm for determining the optimal strategies of the players is proposed. The obtained results for the general positional game model can be used for determining the optimal stationary strategies of the players in the case when a Nash equilibrium exists.

References

1. Condon, A.: The complexity of stochastic games. Inf. Comput. **96(2)**, 203–224 (1992)
2. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. J. Game Theory **8**, 109–113 (1979)
3. Filar, J.A., Vrieze, K.: Competitive Markov Decision Processes. Springer, New York (1997)
4. Gillette, D.: Stochastic games with zero stop probabilities. In: Contribution to the Theory of Games, vol. 3, pp. 179–187. Princeton University Press, Princeton (1957)
5. Gurvich, V.A., Karzanov, A.V., Khachian, L.G.: Cyclic games and an algorithm to find minimax cycle means in directed graphs. USSR Comput. Math. Math. Phys. **28**, 85–91 (1988)
6. Howard, R.A.: Dynamic Programming and Markov Processes. Wiley, New York (1960)
7. Lal, A.K., Sinha S.: Zero-sum two person semi-Markov games. J. Appl. Prob. **29**, 56–72 (1992)
8. Lozovanu, D.: The game-theoretical approach to Markov decision problems and determining Nash equilibria for stochastic positional games. Int. J. Math. Model. Numer. Optim. **2**(2), 162–164 (2011)
9. Lozovanu, D., Pickl, S.: Nash equilibria conditions for cyclic games with p players. Electron. Notes in Discrete Math. **25**, 117–124 (2006)
10. Lozovanu, D., Pickl, S.: Optimization and Multiobjective Control of Time-Discrete Systems. Springer, Berlin (2009)

11. Lozovanu, D., Pickl, S.: Nash equilibria conditions for stochastic positional games. In: Contributions to Game Theory and Management, vol. 7, pp. 201–2013. St. Petersburg State University, St. Petersburg (2014)
12. Lozovanu, D., Pickl, S., Kropat, E.: Markov decision processes and determining Nash equilibria for stochastic positional games. In: Proceedings of 18th World Congress IFAC-2011, pp. 13398–13493 (2011)
13. Mertens, J.F., Neyman, A.: Stochastic games. Int. J. Game Theory **10**, 53–66 (1981)
14. Moulin, H.: Prolongement des jeux a deux joueurs de somme nulle. Bull. Soc. Math. Fr. Mem. **45**, 5–111 (1976)
15. Nash, J.F.: Non cooperative games. Ann. Math. **2**, 286–295 (1951)
16. Neyman, A., Sorin, S.: Stochastic Games and Applications. NATO ASI Series. Kluwer Academic, Dordrecht (2003)
17. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New Jersey (2005)
18. Shapley, L.: Stochastic games. Proc. Natl. Acad. Sci. USA **39**, 1095–1100 (1953)

Adaptive Tunning of All Parameters in a Multi-Swarm Particle Swarm Optimization Algorithm: An Application to the Probabilistic Traveling Salesman Problem

Yannis Marinakis, Magdalene Marinaki, and Athanasios Migdalas

Abstract One of the main issues in the application of a particle swarm optimization (PSO) algorithm and of every evolutionary optimization algorithm is the finding of the suitable parameters of the algorithm. Usually, a trial and error procedure is used but, also, a number of different procedures have been applied in the past. In this chapter, we use a new adaptive version of a PSO algorithm where random values are assigned in the initialization of the algorithm and, then, during the iterations the parameters are optimized together and simultaneously with the optimization of the objective function of the problem. This idea is used for the solution of the probabilistic traveling salesman problem (PTSP). The algorithm is tested on a number of benchmark instances and it is compared with a number of algorithms from the literature.

Keywords Particle swarm optimization • Variable neighborhood search • Probabilistic traveling salesman problem

1 Introduction

Particle Swarm Optimization (PSO) is a population-based swarm intelligence algorithm that was originally proposed by Kennedy and Eberhart [29]. Usually in a PSO algorithm and, in general, in all evolutionary optimization algorithms, a set

Y. Marinakis (✉) • M. Marinaki
School of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece
e-mail: marinakis@ergasya.tuc.gr; magda@dssl.tuc.gr

A. Migdalas
Department of Civil Engineering, Aristotle University of Thessalonike, 54124 Thessalonike, Greece

Industrial Logistics, Luleå University of Technology, 97187 Luleå, Sweden
e-mail: samig@civil.auth.gr; athmig@ltu.se

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_10

of parameters are selected or calculated in the beginning of the algorithm and, then, this set is used for the whole process. These parameters are calculated for some instances, in which they probably give best solutions, and, then, they are used for all the other instances. The questions that arise from this procedure are: "Are these parameters the optimum for every instance of the selected problem?" And if they are not: "Is there an efficient procedure that could estimate the best parameters for all instances?" And the final question: "Is there a set of best parameters for every instance in an optimization problem?" In this chapter, we are trying to answer all these questions. It should be noted that it is almost impossible to have the same parameters for all the instances of the problem but, then, which parameters will we use for the solution of the problem? As it is difficult to optimize the parameters for each instance independently and, then, to use it for the rest of the instances, we present a procedure that uses a simple optimization phase inside the algorithm to optimize the parameters together and simultaneously with the optimization of the objective function. The reason that a classic Constriction PSO algorithm is used is that we would like to test the idea of the optimization of the parameters in the most commonly used version of the PSO algorithm. The algorithm is tested on the probabilistic traveling salesman problem (PTSP) for a number of reasons. The first one is that it is an interesting NP-hard problem, the second is that there are a number of benchmark instances in the literature that could be used for testing the proposed algorithm, the adaptive multi-swarm particle swarm optimization (AMPSO) algorithm, and the third one is that we have published in the past another paper [38] for the same problem with PSO variants and parameters of the algorithms calculated in the beginning of the process and, thus, it would be interesting to make comparisons with these approaches.

The finding of the optimal set of parameters is not a new idea. In the literature and for the various variants of the PSO algorithm, authors have proposed different ways for calculating the main parameters of the algorithms. The most common way is to use the parameters that most researchers have used in the literature. However, this is not the most efficient way as the best parameters for one optimization problem may not necessarily be the best parameters for another optimization problem. A more efficient, but still not totally effective way, especially, when a huge number of instances exist, is to test for some instances a number of different sets of parameters, find the best values of the parameters for these instances, and, then, use these values for the rest of the instances. In this case, there is not any assurance that these values will give the best results for all instances. Nowadays, a number of algorithms have been proposed for automated tunning of the parameters inside the procedure. Most of the papers are using an adaptive way to increase or decrease through the iterations the inertia factor [17, 32, 36, 42, 44, 56, 58, 60, 62], the acceleration coefficients [12, 57], or both [22, 23, 26, 46, 50, 51, 53, 54, 59]. Most of these algorithms are denoted as adaptive particle swarm optimization (APSO) algorithms [59]. In all researches, the equations used to adapt the selected parameters are not the same; however, the main idea is the same. Another way to adapt some parameters (usually the inertia weight) is by using a fuzzy system [1, 28, 43, 45, 47].

In [6] the authors present a variant of the PSO algorithm using a strategy that changes the population size and simultaneously preserves the diversity of the population. The adjustment of the population size is performed automatically according to the value of diversity of the population in ultimate time of current ladder. (The authors divide the generations of the PSO algorithm in equal time periods that are denoted as ladders.) Using grey relational analysis the authors in [30, 31] proposed two grey-based parameter automation strategies for PSO. These two strategies are used, the one in the inertia weight and the other in acceleration coefficients. TRIBES [7] avoids manual tuning by defining adaptation rules which aim at automatically changing the particles, behaviors as well as the topology of the swarm. In TRIBES, the topology is changed according to the swarm behavior and the strategies of displacement are chosen according to the performances of the particles [9, 10]. Apart from the authors who proposed TRIBES, a number of other authors have applied them [48, 52].

In [11] a multilevel approach is used in order to determine the most appropriate set of parameters for the choice function. The parameters are fine-tuned by a PSO algorithm which trains the choice function carrying out a sampling phase. In [27] age-group topology is proposed for the tunning of the parameters and for the adaptation of the population. In order to keep population diversity during searching, the particles are separated to different age groups by their age and particles in each age group can only select the ones in younger groups or their own groups as their neighborhoods. In [61] an APSO with an adaptation strategy for swarm size, inertia factor, and acceleration coefficients is presented. A number of papers that are using Learning Automata with different learning strategies in order to adapt the basic parameters of PSO have been presented [19–21, 55].

In this chapter, a new algorithm, the AMSPSO, is presented where all parameters (acceleration coefficients, iterations, local search iterations, upper and lower bounds of the velocities and of the positions, number of swarms, and number of particles in each swarm) are optimized during the procedure and, thus, the algorithm works independently and without any interference from the user. All parameters are randomly initialized and, afterwards, during the iterations the parameters are adapted using three different conditions: the first is used for all parameters except the number of particles, the second is used for the increase of the number of particles, and the third is used for the decrease of the number of particles.

The rest of the chapter is organized as follows: In the next section a brief description of the PTSP is given while in the third section the proposed algorithm, the AMSPSO algorithm, is presented and analyzed in detail. Computational results are presented and analyzed in the fourth section while in the last section conclusions and future research are given.

2 Probabilistic Traveling Salesman Problem

The problem studied in this chapter is the *Probabilistic Traveling Salesman Problem (PTSP)*. A number of publications concerning the PTSP are given in [3, 24, 25, 49]. In this problem, which is a variant of the traveling salesman problem (TSP), a customer will be present (with probability p) or not (with probability $1-p$) in a specific route during a day. Thus, while in the TSP, a tour with minimum cost should be calculated, in the PTSP the objective is the minimization of the expected length of the a priori tour where each customer requires a visit only with a given probability [38]. The a priori tour is a template for the visiting sequence of all customers. When an instance is needed to be solved, initially, the a priori tour will be calculated and, then, the customers should be visited based on the sequence of the a priori tour while the customers who do not need to be visited will simply be skipped [34]. PTSP is an NP-hard problem [3]. The main formulation of the PTSP can be found in [4, 24]. In this formulation the expected length for the a priori tour $\tau = (1, 2, ..., n)$ is minimized:

$$E[L_\tau] = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} p_i p_j \prod_{k=i+1}^{j-1} (1-p_k) + \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} p_i p_j \prod_{k=i+1}^{n} (1-p_k) \prod_{l=1}^{j-1} (1-p_l) \quad (1)$$

where the length of the tour τ is denoted by L_τ, n are the potential customers ($V = \{1, 2....n\}$), and d_{ij} is the distance between the nodes i and j. For analytical presentation and analysis of the formulation that is used in this chapter please see [38].

3 Adaptive Multi-Swarm Particle Swarm Optimization Algorithm

In the following, an analytical description of the proposed algorithm, the AMSPSO algorithm, is given. In a PSO algorithm, initially, a set of particles is created randomly where each particle corresponds to a possible solution. Each particle has a position in the space of solutions and moves with a given velocity. Each particle is recorded via the path representation of the tour, that is, via the specific sequence of the nodes. As the calculation of the velocity of each particle is performed by Eq. (2) (see below), the above-mentioned representation should be transformed appropriately. We transform each element of the solution into a floating point in the interval (0,1], calculate the velocities and the positions of all particles and, then, convert back the particles' positions into the integer domain using relative position indexing [33].

The position of each particle is represented by a d-dimensional vector in problem space $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, $i = 1, 2, \ldots, N$ (N is the population size and d is the number of the vector's dimension), and its performance is evaluated on the predefined fitness function. The velocity v_{ij} represents the changes that will be made to move the particle from one position to another. Three possible directions a particle can follow: the particle can follow its own path, it can move towards the best position it had during the iterations ($pbest_{ij}$), or it can move to the best particle's position ($gbest_j$). The velocity and position equations are updated as follows (constriction PSO) [8]:

$$v_{ij}(t+1) = \chi(v_{ij}(t) + c_1 rand_1(pbest_{ij} - x_{ij}(t)) + c_2 rand_2(gbest_j - x_{ij}(t))) \tag{2}$$

and

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{3}$$

where

$$\chi = \frac{2}{|2 - c - \sqrt{c^2 - 4c}|} \text{ and } c = c_1 + c_2, c > 4 \tag{4}$$

t is the iterations counter, c_1 and c_2 are the acceleration coefficients, $rand_1$ and $rand_2$ are two random variables in the interval (0, 1). A modified version of a local search strategy based on the variable neighborhood search (VNS) algorithm [18] is applied [39] in each particle in the swarm in order to improve the solutions produced from the PSO algorithm. Finally, a modified version of a Path Relinking strategy [16] with starting solution the best particle and target solution one of the other particles of the swarm is applied [39]. In each iteration of the algorithm, the best solution of the whole swarm and the best solution of each particle are kept.

The most important and novel part of the algorithm is the optimization of the parameters during the iterations of the algorithm. Initially, random values of the parameters are given taking into consideration the fact that these values should not violate some specific bounds. For example, the sum of c_1 and c_2 should be greater than 4 due to Eq. (4). The parameters that should be optimized are the number of swarms, the number of particles, the number of iterations, the number of local search iterations, the c_1 and c_2, and the upper and lower bounds in positions and in velocities. In order to have less values to optimize, the lower bounds in positions and velocities are always set equal to the negative values of the upper bounds in positions and velocities, respectively. The upper bounds could not have a value less than a specific value, in this chapter this value is selected equal to 0.1, as if the upper bounds take negative values, then the PSO algorithm could not work properly. Another value that was selected as a threshold value is a value corresponding to the number of consecutive iterations (*Max iterations*) with no improvement in the results of the best solution (this value is selected to be equal with the initial number of particles).

After the initialization of the particles, the fitness function of each particle is calculated. The initial velocities are set equal to zero. Also, the average values in the fitness function of all particles and the average values of the best solutions of all particles are calculated too. In the first iteration, these average values are equal for all particles. The initial random values of all parameters are the best values so far, and the algorithm proceeds as a constriction PSO. The best values for each parameter are updated when the average values of the best solutions of all particles in a specific iteration are improved. Three different conditions are controlling the parameters during the iterations of the algorithm. In the first one, if for a consecutive number of iterations the best solution has not been improved, the values of the number of local search iterations, the c_1 and c_2, and the upper and lower bounds in positions and in velocities are updated as follows:

$$c_1 = c_{1opt} + \alpha \tag{5}$$

$$c_2 = c_{2opt} + \alpha_1 \tag{6}$$

$$u_{positions} = UB + \alpha_2 \tag{7}$$

$$l_{positions} = -u_{positions} \tag{8}$$

$$u_{velocities} = V + \alpha_3 \tag{9}$$

$$l_{velocities} = -u_{velocities} \tag{10}$$

$$Local\ Search\ iter = LS + \alpha_4 \tag{11}$$

where c_{1opt}, c_{2opt}, UB, V, and LS are the best values for the c_1, c_2, upper bounds in positions, upper bounds in velocities, and local search iterations, respectively, and $\alpha, \alpha_1, \alpha_2, \alpha_3,$ and α_4 are calculated from the following equations:

$$\alpha = \frac{c_1 - c_{1opt}}{c_{1opt}} \tag{12}$$

$$\alpha_1 = \frac{c_2 - c_{2opt}}{c_{2opt}} \tag{13}$$

$$\alpha_2 = \frac{u_{positions} - UB}{UB} \tag{14}$$

$$\alpha_3 = \frac{l_{positions} - V}{V} \tag{15}$$

$$\alpha_4 = \frac{Local\ Search\ iter - LS}{LS} \tag{16}$$

and, thus, for all these parameters, if the value of the parameter in the current iteration (e.g., the c_1 value) is less than the best value of the corresponding parameter (e.g., the c_{1opt} value), then the parameter is reduced, otherwise it is increased. In the second condition, the increase of the number of particles and of the number of swarms is performed. If for a consecutive number of iterations, the best solution,

the average best solution of all particles, and of all swarms are not improved, then a number of new particles are created. All new particles are initialized from random values and create a new swarm and they use as initial values of the parameters the current values. The reason that a number of new particles are created is that probably the old particles have stuck in a local optimum and the new particles will, probably, give to the swarm the opportunity to escape from this local optimum. The main advantage of the new particles is that they begin with different values in all parameters and, thus, they have the possibility to search in a completely different place in the solution space. The reason that we use, except of the improvement of the best solution, the average solutions of all particles is that it is probable that all the best solutions of the particles in this iteration are not improved and, thus, the possibility of finding a better best solution is decreased. The increase of the number of particles is performed using the following equation:

$$Particles = NP + \alpha_5 \tag{17}$$

where NP is the optimum value of particles and α_5 is calculated from the following equation:

$$\alpha_5 = \frac{Particles - NP}{NP} \tag{18}$$

Finally, in the third condition, the decrease of the number of particles is performed. If for a consecutive number of iterations the best solution has not been improved and the best value of a particle is more than 5 % of the best value of the swarm, then this particle is deleted from the swarm. Also, if for the whole swarm, the same condition is hold, then the whole swarm is deleted. The decrease of the particles is necessary for two reasons. Initially, if we have only the increase of the particles, the number of particles, probably, after a number of iterations will lead to a very difficult to handle number of particles. Second, a particle with bad solution will, probably, not give anything more in the swarm and only will delay the convergence of the algorithm. The number of consecutive iterations was set equal to $abs(Initial\ Number\ of\ Particles - Particles)$ if the number of particles has been changed at least one time during the iterations, otherwise the value was set equal to $Initial\ Number\ of\ Particles - \frac{Max\ iterations}{abs(Max\ iterations - Local\ Search\ iter}$). The reason that the absolute value of the difference was used is because there is a possibility to increase or to decrease the number of particles and this value is needed to be always positive. We increase and decrease the number of particles during an iteration as we would like to take into advantage all the possibilities of replacing bad solutions with new probably good ones and to explore different places in the solution space. When the algorithm converges, except of the optimum solution, the optimum parameters have, also, been calculated.

4 Computational Results

All algorithms were implemented in modern Fortran and compiled with the Lahey f95 compiler. PTSP instances were generated starting from TSP instances and assigning to each customer a probability p of requiring a visit. The test instances were taken from the TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95). The algorithm was tested on a set of Euclidean sample problems with sizes ranging from 51 to 1,400 nodes. The instances in all tables are described with their name and the size which means that the instance Pr1002 has 1,002 nodes. The efficiency of the AMSPSO algorithm is calculated using the quality ω of the solutions ($\omega = \frac{(c_{AMSPSO} - c_{BKS})}{c_{BKS}}\%$, where c_{AMSPSO} is the cost of the solution produced by AMSPSO and c_{BKS} is the cost of the best known solution (BKS)). To test the performance of the proposed algorithm we applied AMSPSO (and the other algorithms used in the comparisons) ten times to each test instance.

In Table 1, the most important results of the proposed algorithm are presented analytically. The table is divided into three parts. In the first part the results of three instances (eil51, kroA100 and eil101) are presented, while in the second and third parts the result of the other six instances (d198, pr439, and p654 for the second part; rat783, pr1002, and fl1400 for the third part) are presented. For each instance the results concern the quality of the solutions, the optimization of the parameters of the AMSPSO, and the average values of the ten runs. All results of the tables, except the average values of the ten runs, are referring to the best run out of the ten runs for each instance.

More precisely, in the first row (denoted by BKS) of each part of Table 1, the BKS from the literature is presented while in the second row the solution of the best run (denoted by *AMSPSO*) of the proposed algorithm is given and in the third row the quality of this solution (denoted by ω) is presented. In the next eight rows of each part of the table, the results concerning the optimization of the parameters for the best run of the algorithm are presented. More precisely, in row three the iteration number (*Iter*) where the algorithm converged for each set is presented while in the fourth row the optimized number of particles (*NP*) needed for each instance is presented. When we mention that the optimized number of a parameter, i.e., particles, is presented we do not mean that this number is the optimum number of the parameter but we mean that the algorithm converged to this number and using this number the algorithm gave the best results. However, this number was not used in all iterations of the algorithm but only when it converged to this number. This means that if we begin the algorithm from random solutions using the optimized (best) parameters found during the iterations, the algorithm will probably find a different solution than the one produced with the procedure of the proposed algorithm. This was expected as the reason that we proposed this algorithm is to avoid to search before the beginning of iterations for a set of good parameters and to succeed to converge to a set, different in each instance and in each run, during the iterations of the algorithm without the interference of the user in any stage of the algorithm.

Table 1 Analytical presentation of the results of the proposed algorithm

	eil51			kroA100			eil101		
	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
BKS	130.12	310.75	407.92	9034	16569	20508.77	196	455.65	601.5
AMSPSO	129.42	310.75	407.92	9034.97	16569.7	20508.77	197.37	455.65	601.5
ω	−0.54	0.00	0.00	0.01	0.00	0.00	0.70	0.00	0.00
Iter	49	100	99	117	138	99	112	82	121
NP	49	103	51	76	86	124	118	96	79
c_{1opt}	2.28	2.66	2.48	2.64	2.28	2.56	2.83	2.60	2.79
c_{2opt}	2.49	2.69	2.45	2.76	2.28	2.46	2.75	2.56	2.67
UB	3.30	2.95	2.81	3.52	3.15	2.47	2.48	3.10	2.64
V	4.21	4.17	5.21	5.51	4.25	3.94	6.25	5.36	4.62
LS	97	129	109	106	71	143	150	150	150
Swarm	7	2	2	5	2	2	1	6	6
AvPar	138.95	321.82	414.79	9045.87	16583.92	20523.28	208.15	473.03	613.74
ω_{AvPar}	6.79	3.56	1.68	0.13	0.09	0.07	6.20	3.81	2.03
AvOpt	133.24	315.73	412.36	9039.92	16571.85	20513.93	203.26	458.34	606.57
ω_{AvOpt}	2.39	1.60	1.09	0.07	0.02	0.03	3.71	0.59	0.84
	d198			pr439			p654		
	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
BKS	7437	12417	15210.4	49926.89	88728.37	104584.67	19663.66	28383.71	33631.26
AMSPSO	7437.24	12438.54	15210.4	49926.89	88728.37	104584.67	19663.66	28383.71	33631.26
ω	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Iter	163	155	142	139	154	223	134	173	158
NP	73	95	84	82	96	86	95	108	128
c_{1opt}	2.40	2.62	2.86	2.86	2.78	2.66	2.60	2.76	2.86
c_{2opt}	2.52	2.65	2.80	2.89	2.86	2.60	2.68	2.59	2.91
UB	4.05	3.59	2.82	1.98	2.72	3.38	2.55	3.54	2.34

(continued)

Table 1 (continued)

	d198			pr439			p654		
	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
V	5.65	5.94	5.44	4.80	5.94	5.90	4.83	5.76	5.44
LS	111	141	142	146	150	149	150	149	149
Swarm	5	1	3	7	5	3	8	2	7
AvPar	7455.27	12463.42	15229.08	49955.39	88754.08	104610.69	19689.31	28396.19	33655.55
ω_{AvPar}	0.25	0.37	0.12	0.06	0.03	0.02	0.13	0.04	0.07
AvOpt	7442.57	12444.40	15212.64	49933.44	88732.69	104590.05	19667.56	28389.60	33635.65
ω_{AvOpt}	0.07	0.22	0.01	0.01	0.00	0.01	0.02	0.02	0.01
	rat783			pr1002			fl1400		
	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
BKS	3246	6846	8604.28	110764.625	207210.53	254365.75	9689.55	16416.11	19679.24
AMSPSO	3312.15	6924.51	8592.21	110764.625	207210.53	254365.75	9689.55	16416.11	19679.24
ω	2.04	1.15	−0.14	0.00	0.00	0.00	0.00	0.00	0.00
Iter	123	104	180	101	199	112	149	144	158
NP	102	98	126	91	154	121	96	92	42
c_{1opt}	2.86	2.90	2.66	2.65	2.89	2.61	2.74	2.35	2.40
c_{2opt}	2.78	2.87	2.42	2.55	2.75	2.96	2.80	2.53	2.50
UB	2.83	3.22	2.88	2.91	2.19	2.58	3.43	3.60	2.90
V	4.58	5.10	4.62	5.08	5.89	6.42	6.02	5.03	5.35
LS	148	147	118	101	147	149	137	110	93
Swarm	7	6	2	4	1	6	2	3	2
AvPar	3326.97	6939.50	8598.43	110783.56	207232.76	254377.95	9704.42	16439.88	19706.97
ω_{AvPar}	2.49	1.37	−0.07	0.02	0.01	0.00	0.15	0.14	0.14
AvOpt	3316.11	6926.65	8595.69	110766.76	207215.75	254368.08	9692.89	16420.29	19685.85
ω_{AvOpt}	2.16	1.18	−0.10	0.00	0.00	0.00	0.03	0.03	0.03

In the sixth and seventh rows of each part of the table, the optimized values of c_1 and c_2 (c_{1opt} and c_{2opt}) are presented, respectively, while in the next two rows the optimized values for the upper bound (*UB*) for the positions and the velocities (*V*) are presented, respectively. The lower bounds for the positions and for the velocities are the negative values of the upper bounds. In rows ten and eleven, the optimized number of local search (*LS*) iterations and of the number of swarms (*Swarm*) are presented. Finally, in the last four rows of the table, the average values of the ten runs are presented. More precisely, initially the average solution of each particle (*AvPar*) and then its quality (ω_{AvPar}) are presented and finally the average best solution of each particle (*AvOpt*) and its quality are given (ω_{AvOpt}).

Initially, we have to mention that each instance was tested with three different probability values (0.1, 0.5, and 0.9) and, thus, the total number of different instances instead of 9 is 27. The proposed algorithm found new best solution in two instances, in the eil51 with probability equal to 0.1 and in the rat783 with probability equal to 0.9. In the other 20 instances, the proposed algorithm found the BKS from the literature. In three instances, the quality of the solution is between 0.01 and 0.70 and in, only, two instances the quality of the solutions is larger than 1 (in rat783 with probability 0.5 the quality of the solution is equal to 1.15 and in the same instance with probability 0.1 the quality of the solution is equal to 2.04). Simultaneously, the average of ten runs of the algorithm shows that the proposed methodology is very efficient. This is proved by the fact that in the average of ten runs, with different parameters as in each run the algorithm converged in different parameters (the parameters are optimized for the specific run), the algorithm found in 1 instance average quality less than zero, in 4 instances average quality equal to zero, in 13 instances average quality between 0.01 and 0.10, in 3 instances between 0.22 and 0.89, and, only, in 6 instances the algorithm found average quality larger than 1.

The most important values of Table 1 are the values where the optimized parameters for each of the main parameters of the algorithm are presented. Usually, these parameters are given in the initial phase of the algorithm and, then, are constant for all the iterations of the algorithm. The novelty of the proposed algorithm is that, initially, a random value for each of these parameters is given and, then, an optimization of each of these parameters is performed inside the algorithm. As it can be seen, the convergence of the algorithm is succeeded using between 49 iterations and 223 iterations. The optimum number of particles varies between 49 and 128. The c_{1opt} and the c_{2opt} vary between 2.28 and 2.86 and between 2.28 and 2.89, respectively. The upper bounds of the positions and the velocities vary between 1.98 and 4.05 and between 3.94 and 6.42, respectively. The values of the local search iterations vary between 71 and 150. Finally, the number of swarms varies between 1 and 8. These variations of the values are very important as they indicate that for two different instances the most probable is that the algorithm needs at least one different parameter (possibly more than one different parameters) to find its best solution. Thus, the optimization of the parameters in addition to the optimization of the objective function improves the effectiveness of the AMSPSO algorithm.

In Table 2, a comparison of the proposed algorithm with three other versions of PSO is presented. More precisely, the algorithms used in these comparisons are a classic PSO algorithm with one swarm and a simple local search, a more efficient PSO algorithm denoted as HybPSO with a more advanced local search algorithm, and one swarm and a multi-swarm PSO (HybMSPSO) algorithm. The HybPSO algorithm was first used for the solution of the Vehicle Routing Problem [41] and the HybMSPSO algorithm was first used for the solution of the same problem studied in this chapter, the PTSP [38]. For analytical presentation and analysis of these algorithms please see [38, 41]. The comparison of the results of the proposed algorithm with the results of these three algorithms, especially the HybMSPSO, are very important as there are three algorithms that have been applied in the past [38] in the same problem with remarkable results, the HybMSPSO is one of the most efficient algorithms published for the PTSP, and they are all algorithms that use a variant of PSO algorithm. Moreover, the HybMSPSO algorithm is a multi-swarm PSO algorithm as the proposed one. The parameters of HybMSPSO optimized off-line before the procedure starts and they are the same for all instances. Thus, in this table, initially, the results of the BKS from the literature for every probability are given, and, then, in the next eight columns the solutions and the quality of the solutions of PSO algorithm, HybPSO algorithm, HybMSPSO algorithm, and of the proposed AMSPSO algorithm are presented. If we compare the results of the proposed algorithm with the results of PSO algorithm, the proposed algorithm finds better solutions in 29 instances and the two algorithms find the same solutions in 1 instance. When we compare the results of the proposed algorithm with the results of HybPSO algorithm, the proposed algorithm finds better solutions in 25 instances and the two algorithms find the same solutions in 5 instances. Finally, the comparison of the results of the proposed algorithm with the results of HybMSPSO shows that the proposed algorithm outperforms the HybMSPSO in 20 instances and in the other 10 instances the two algorithms find the same solutions. Thus, as both algorithms are multi-swarm PSO algorithms the use of the automated tunning of the parameters in the proposed algorithm gave a more effective algorithm than the one where the parameters are optimized before the procedure begins. Thus, with the use of the automated tunning of the parameters the algorithm gave better results and we avoided the procedure of finding the suitable parameters before testing all instances.

In Table 3, a comparison of the proposed algorithm with three other algorithms is presented. The first one is an implementation of a classic Tabu Search [14, 15], the second one is a variant of GRASP algorithm [13], the ENS-GRASP [40], and the last one is a variant of Honey Bees Mating Optimization algorithm, the HBMOPTSP [37]. The way these three algorithms are applied in the PTSP and the results produced are presented and analyzed in [40] for the first two algorithms and in [37] for the third one. The comparison of the results of these three algorithms with the results of the proposed algorithm is important in order to see what is the effectiveness of the proposed algorithm in relation with two very powerful metaheuristics, as the Tabu Search and the GRASP and one very efficient nature inspired algorithm as the HBMOPTSP algorithm. The structure of the table is the

Table 2 Comparison of the results of the proposed algorithm with the results of other versions of PSO

			PSO		HybPSO		HybMSPSO		AMSPSO	
		BKS	Cost	ω	Cost	ω	Cost	ω	Cost	ω
eil51	0.1	130.12	130.26	0.11	130.17	0.04	130.12	0.00	129.42	−0.54
	0.5	310.75	312.18	0.46	311.04	0.09	310.75	0.00	310.75	0.00
	0.9	407.92	408.21	0.07	407.92	0.00	407.92	0.00	407.92	0.00
kroA100	0.1	9034	9076.23	0.47	9074.94	0.45	9074.94	0.45	9034.97	0.01
	0.5	16569	16584.32	0.09	16584.32	0.09	16581.64	0.08	16569.70	0.00
	0.9	20508.77	20508.77	0.00	20508.77	0.00	20508.77	0.00	20508.77	0.00
eil101	0.1	196	201.01	2.56	200.03	2.06	200.03	2.06	197.37	0.70
	0.5	455.65	457.23	0.35	455.73	0.02	455.65	0.00	455.65	0.00
	0.9	601.5	602.21	0.12	601.50	0.00	601.50	0.00	601.50	0.00
ch150	0.1	2479	2515.08	1.46	2514.31	1.42	2510.11	1.25	2482.54	0.14
	0.5	5004	5017.23	0.26	5016.85	0.26	5016.85	0.26	5016.14	0.24
	0.9	6292.01	6293.23	0.02	6292.01	0.00	6292.01	0.00	6292.01	0.00
d198	0.1	7437	7512.12	1.01	7504.94	0.91	7504.94	0.91	7437.24	0.00
	0.5	12417	12537.26	0.97	12534.92	0.95	12527.56	0.89	12438.54	0.17
	0.9	15210.4	15225.28	0.10	15224.18	0.09	15216.61	0.04	15210.40	0.00

(continued)

Table 2 (continued)

			PSO		HybPSO		HybMSPSO		AMSPSO	
		BKS	Cost	ω	Cost	ω	Cost	ω	Cost	ω
pr439	0.1	49926.89	50408.99	0.97	50373.95	0.90	50066.27	0.28	49926.89	0.00
	0.5	88728.37	90121.76	1.57	88841.09	0.13	88752.34	0.03	88728.37	0.00
	0.9	104584.7	104742.28	0.15	104656.70	0.07	104584.67	0.00	104584.67	0.00
p654	0.1	19663.66	19821.01	0.80	19728.54	0.33	19690.95	0.14	19663.66	0.00
	0.5	28383.71	28401.45	0.06	28383.71	0.00	28383.71	0.00	28383.71	0.00
	0.9	33631.26	33649.33	0.05	33642.60	0.03	33631.26	0.00	33631.26	0.00
rat783	0.1	3246	3621.21	11.56	3618.01	11.46	3616.44	11.41	3312.15	2.04
	0.5	6846	7097.85	3.68	7097.33	3.67	7094.87	3.64	6924.51	1.15
	0.9	8604.28	8625.26	0.24	8625.26	0.24	8604.28	0.00	8592.21	−0.14
pr1002	0.1	110764.6	111888.21	1.01	111547.03	0.71	110873.41	0.10	110764.63	0.00
	0.5	207210.5	209330.70	1.02	207916.81	0.34	207605.50	0.19	207210.53	0.00
	0.9	254365.8	254830.50	0.18	254819.95	0.18	254593.28	0.09	254365.75	0.00
fl1400	0.1	9689.55	9752.24	0.65	9727.04	0.39	9703.32	0.14	9689.55	0.00
	0.5	16416.11	16544.93	0.78	16541.93	0.77	16416.11	0.00	16416.11	0.00
	0.9	19679.24	19732.23	0.27	19729.55	0.26	19729.55	0.26	19679.24	0.00

Table 3 Comparison of the results of the proposed algorithm with the results of other algorithms

		Tabu search		ENS-GRASP		HBMOPTSP		AMSPSO	
		Cost	ω	Cost	ω	Cost	ω	Cost	ω
eil51	0.1	130.82	0.54	130.30	0.14	130.12	0.00	129.42	−0.54
	0.5	313.50	0.88	311.04	0.09	310.75	0.00	310.75	0.00
	0.9	411.63	0.91	407.92	0.00	407.92	0.00	407.92	0.00
kroA100	0.1	9116.64	0.91	9079.86	0.51	9071.72	0.42	9034.97	0.01
	0.5	16658.46	0.54	16584.32	0.09	16581.64	0.08	16569.70	0.00
	0.9	20510.21	0.01	20508.77	0.00	20508.77	0.00	20508.77	0.00
eil101	0.1	202.42	3.28	200.03	2.06	200.03	2.06	197.37	0.70
	0.5	461.52	1.29	455.73	0.02	455.65	0.00	455.65	0.00
	0.9	602.35	0.14	601.51	0.00	601.50	0.00	601.50	0.00
ch150	0.1	2554.59	3.05	2520.10	1.66	2509.98	1.25	2482.54	0.14
	0.5	5071.51	1.35	5016.85	0.26	5016.82	0.26	5016.14	0.24
	0.9	6294.32	0.04	6292.01	0.00	6292.01	0.00	6292.01	0.00
d198	0.1	7525.03	1.18	7525.03	1.18	7490.09	0.71	7437.24	0.00
	0.5	12606.23	1.52	12538.45	0.98	12492.62	0.61	12438.54	0.17
	0.9	15227.34	0.11	15225.26	0.10	15210.40	0.00	15210.40	0.00
pr439	0.1	50848.29	1.85	50402.09	0.95	49926.89	0.00	49926.89	0.00
	0.5	90463.31	1.96	88914.76	0.21	88728.37	0.00	88728.37	0.00
	0.9	104828.21	0.23	104735.27	0.14	104584.67	0.00	104584.67	0.00
p654	0.1	20034.61	1.89	19766.74	0.52	19663.66	0.00	19663.66	0.00
	0.5	28510.05	0.45	28388.01	0.02	28383.71	0.00	28383.71	0.00
	0.9	33737.12	0.31	33646.79	0.05	33631.26	0.00	33631.26	0.00
rat783	0.1	3705.31	14.15	3618.01	11.46	3616.44	11.41	3312.15	2.04
	0.5	7123.76	4.06	7097.85	3.68	7085.48	3.50	6924.51	1.15
	0.9	8677.34	0.85	8625.26	0.24	8604.28	0.00	8592.21	−0.14
pr1002	0.1	113868.22	2.80	111959.65	1.08	110764.63	0.00	110764.63	0.00
	0.5	210639.29	1.65	207916.81	0.34	207210.53	0.00	207210.53	0.00
	0.9	255001.23	0.25	254819.95	0.18	254365.75	0.00	254365.75	0.00
fl1400	0.1	9767.43	0.80	9727.04	0.39	9689.55	0.00	9689.55	0.00
	0.5	16570.99	0.94	16541.93	0.77	16416.11	0.00	16416.11	0.00
	0.9	19857.28	0.90	19729.55	0.26	19679.24	0.00	19679.24	0.00

same with the structure of Table 3. If we compare the results of the proposed algorithm with the results of Tabu Search algorithm, the proposed algorithm finds better solutions in all 30 instances. When we compare the results of the proposed algorithm with the results of ENS-GRASP algorithm, the proposed algorithm finds better solutions in 26 instances and the two algorithms find the same solutions in 4 instances. Finally, the comparison of the results of the proposed algorithm with the results of HBMOPTSP shows that the proposed algorithm outperforms the HybMSPSO in 11 instances and in the other 19 instances the two algorithms find the same solutions.

The results of the algorithm are, also, compared (Table 4) with the results of a number of implementations of ant colony optimization (ACO) metaheuristic taken from [2, 4], and [5], with the results of a number of estimation-based metaheuristics (ILS-EE, MAGX-EE, ACS-EE) proposed by Balaprakash et al. [2] and with the results of a number of evolutionary algorithms with different generators of the initial solutions (RAN, NN1, and NN2) [35]. Also, in this table, for completeness reasons, the results of the two most efficient algorithms of the two previous tables (Tables 2 and 3) are given. In this table, if from a specific instance there are not any values in the table it means that in the paper that the specific algorithm was published the authors did not run their algorithm for this specific instance. Thus, the proposed algorithm performs better from the ACO and pACS algorithms in all instances, in 18 out of 20 instances from RAN (in the other two the RAN algorithm performs better), in 7 out of 13 instances from ILS-EE (in the other 6 instances, the ILS-EE algorithm performs better), in 6 out of 13 instances from MAGX-EE (in the other 7 instances, the MAGX-EE algorithm performs better), in 5 out of 13 instances from ACS-EE (in the other 8 instances, the ACS-EE algorithm performs better), in 4 out of 13 instances from pACS+1-shift (in the other 9 instances, the pACS+1-shift algorithm performs better), in 11 out of 20 instances from NN1 (in 1 instance the two algorithms find the same results and in the other 8 instances, the NN1 algorithm performs better), and in 14 out of 20 instances from NN2 (in 2 instances, the two algorithms find the same results and in the four instances, the NN2 algorithm performs better). In general, none of all these algorithms perform better in all instances. However, the proposed algorithm is a competitive algorithm and gives better results in some instances (in most cases more than 50 % of the testing instances) from all algorithms.

5 Conclusions

In this chapter, a new hybridized algorithm based on PSO with adaptive selection of parameters for the solution of the PTSP has been proposed. The resulting hybrid algorithm was tested on a set of benchmark instances and gave new best solutions in a number of them. The algorithm was compared with a number of PSO implementations for the same problem and gave better results. Also, the algorithm was compared with other metaheuristic, nature-inspired and evolutionary algorithms from the literature and gave competitive results with the best of them and better results from all the others. Our future research will be focused on the application of this algorithm to other difficult stochastic routing problems.

Table 4 Comparison of the results of the proposed algorithm with the results of other algorithms from the literature

	AMSPSO		ACO		pACS		HybMSPSO		HBMOPTSP		RAN	
PTSP	Cost	ω	Cost	ω	Cost	ω	Cost	ω	Cost	ω	Cost	ω
kroA100–0.1	9034.97	0.01	–	–	9039.40	0.06	9074.94	0.45	9071.72	0.42	9034.97	0.01
kroA100–0.2	11714.98	0.01	–	–	11720.60	0.06	11726.50	0.11	–	–	11715.14	0.01
kroA100–0.5	16569.70	0.00	–	–	16605.40	0.22	16581.64	0.08	16581.64	0.08	16569.72	0.00
kroA100–0.75	19378.15	0.06	–	–	–	–	20281.91	4.73	–	–	20001.49	3.28
eil101–0.1	197.37	0.70	–	–	199.70	1.89	200.03	2.06	200.03	2.06	197.34	0.68
eil101–0.2	283.85	0.05	–	–	286.70	1.05	284.93	0.43	–	–	283.77	0.02
eil101–0.5	455.65	0.00	460.56	1.08	460.70	1.11	455.65	0.00	455.65	0.00	459.16	0.77
eil101–0.75	562.54	0.27	564.04	0.54	–	–	579.24	3.25	–	–	566.79	1.03
ch150–0.1	2482.54	0.14	–	–	2493.60	0.59	2510.11	1.25	2509.98	1.25	2479.96	0.04
ch150–0.2	3417.25	0.01	–	–	3444.20	0.80	3444.61	0.81	–	–	3439.42	0.66
ch150–0.5	5016.14	0.24	–	–	5051.30	0.95	5016.85	0.26	5016.82	0.26	5134.73	2.61
ch150–0.75	5912.37	0.23	–	–	–	–	5988.34	1.52	–	–	6142.50	4.13
d198–0.1	7437.24	0.00	–	–	7556.10	1.60	7504.94	0.91	7490.09	0.71	7440.58	0.05
d198–0.2	9313.35	0.01	–	–	9489.20	1.90	9415.08	1.11	–	–	9319.35	0.08
d198–0.5	12438.54	0.17	–	–	12613.30	1.58	12527.56	0.89	12492.62	0.61	12687.59	2.18
d198–0.75	14512.87	0.23	–	–	–	–	14876.35	2.74	–	–	14592.21	0.78
rat783–0.1	3312.15	2.04	–	–	3368.90	3.79	3616.44	11.41	3616.44	11.41	3326.46	2.48
rat783–0.2	4625.19	1.59	–	–	4781.20	5.01	4775.14	4.88	–	–	4766.55	4.69
rat783–0.5	6924.51	1.15	–	–	7261.40	6.07	7094.87	3.64	7085.48	3.50	7352.30	7.40
rat783–0.75	8217.31	0.00	–	–	–	–	8217.31	0.00	–	–	8643.55	5.19

(continued)

Table 4 (continued)

	ILS-EE		MAGX-EE		ACS-EE		pACS+1-shift		NN1		NN2	
PTSP	Cost	ω	Cost	ω	Cost	ω	Cost	ω	Cost	ω	Cost	ω
kroA100-0.1	9042.00	0.09	9041.00	0.08	9036.00	0.02	9034.00	0.00	9034.97	0.01	9034.97	0.01
kroA100-0.2	11718.00	0.03	11717.00	0.03	11716.00	0.02	11714.00	0.00	11715.14	0.01	11715.14	0.01
kroA100-0.5	16576.00	0.04	16569.00	0.00	16569.00	0.00	16569.00	0.00	16569.72	0.00	16610.07	0.25
kroA100-0.75	–	–	–	–	–	–	–	–	19366.01	0.00	19791.83	2.20
eil101-0.1	197.00	0.51	197.00	0.51	197.00	0.51	196.00	0.00	197.34	0.68	197.34	0.68
eil101-0.2	–	–	–	–	–	–	–	–	284.43	0.25	283.72	0.00
eil101-0.5	–	–	–	–	–	–	–	–	459.57	0.86	460.07	0.97
eil101-0.75	–	–	–	–	–	–	–	–	561.01	0.00	7563.44	0.43
ch150-0.1	2484.00	0.20	2483.00	0.16	2481.00	0.08	2479.00	0.00	2479.96	0.04	2479.96	0.04
ch150-0.2	3423.00	0.18	3420.00	0.09	3432.00	0.44	3417.00	0.00	3418.95	0.06	3419.36	0.07
ch150-0.5	5005.00	0.02	5005.00	0.02	5004.00	0.00	5033.00	0.58	5012.20	0.16	5016.08	0.24
ch150-0.75	–	–	–	–	–	–	–	–	5898.65	0.00	6005.01	1.80
d198-0.1	7444.00	0.09	7443.00	0.08	7445.00	0.11	7437.00	0.00	7437.99	0.01	7438.47	0.02
d198-0.2	9323.00	0.12	9318.00	0.06	9322.00	0.11	9312.00	0.00	9324.12	0.13	9313.35	0.01
d198-0.5	12424.00	0.06	12421.00	0.03	12417.00	0.00	12464.00	0.38	12580.28	1.31	12559.34	1.15
d198-0.75	–	–	–	–	–	–	–	–	14479.19	0.00	14588.90	0.76
rat783-0.1	3250.00	0.12	3246.00	0.00	3251.00	0.15	3341.00	2.93	3300.56	1.68	3323.81	2.40
rat783-0.2	4556.00	0.07	4553.00	0.00	4570.00	0.37	4826.00	6.00	4725.28	3.78	4738.52	4.07
rat783-0.5	6857.00	0.16	6846.00	0.00	6854.00	0.12	7789.00	13.77	7311.82	6.80	7364.89	7.58
rat783-0.75	–	–	–	–	–	–	–	–	8626.38	4.98	8607.46	4.75

References

1. Bahmani-Firouzi, B., Farjah, E., Azizipanah-Abarghooee, R.: An efficient scenario-based and fuzzy self-adaptive learning particle swarm optimization approach for dynamic economic emission dispatch considering load and wind power uncertainties. Energy **50**, 232–244 (2013)
2. Balaprakash, P., Birattari, M., Stutzle, T., Yuan, Z., Dorigo, M.: Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem. Swarm Intell. **3**, 223–242 (2009)
3. Bertsimas, D.J.: Probabilistic combinatorial optimization problems. Ph.D. thesis, MIT, Cambridge (1988)
4. Bianchi, L.: Ant colony optimization and local search for the probabilistic traveling salesman problem: a case study in stochastic combinatorial optimization. Ph.D. thesis, Universite Libre de Bruxelles, Belgium (2006)
5. Branke, J., Guntsch, M.: Solving the probabilistic TSP with ant colony optimization. J. Math. Model. Algorithms **3**(4), 403–425 (2004)
6. Chen, D.B., Zhao, C.X.: Particle swarm optimization with adaptive population size and its application. Appl. Soft Comput. **9**, 39–48 (2009)
7. Clerc, M.: Particle Swarm Optimization. Wiley-ISTE, London (2006)
8. Clerc, M., Kennedy, J.: The particle swarm: explosion, stability and convergence in a multi-dimensional complex space. IEEE Trans. Evol. Comput. **6**, 58–73 (2002)
9. Cooren, Y., Clerc, M., Siarry, P.: Initialization and displacement of the particles in TRIBES, a parameter-free particle swarm optimization algorithm. In: Cotta, C., et al. (eds.) Adaptive and Multilevel Metaheuristics, SCI, vol. 136, pp. 199–219. Springer, Berlin/Heidelberg (2008)
10. Cooren, Y., Clerc, M., Siarry, P.: Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm. Swarm Intell. **3**, 149–178 (2009)
11. Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., Paredesm, F.: Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. Expert Syst. Appl. **40**, 1690–1695
12. Eslami, M., Shareef, H., Taha, M.R., Khajehzadeh, M.: Adaptive particle swarm optimization for simultaneous design of UPFC damping controllers. Electr. Power Energy Syst. **57**, 116–128 (2014)
13. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedure. J. Global Optim. **6**, 109–133 (1995)
14. Glover, F.: Tabu search I. ORSA J. Comput. **1**(3), 190–206 (1989)
15. Glover, F.: Tabu search II. ORSA J. Comput. **2**(1), 4–32 (1990)
16. Glover, F., Laguna, M., Marti, R.: Scatter search and path relinking: advances and applications. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 1–36. Kluwer Academic Publishers, Boston (2003)
17. Han, F., Ling, Q.H.: A new approach for function approximation incorporating adaptive particle swarm optimization and a priori information. Appl. Math. Comput. **205**, 792–798 (2008)
18. Hansen, P., Mladenovic, N.: Variable neighborhood search: principles and applications. Eur. J. Oper. Res. **130**, 449–467 (2001)
19. Hasanzadeh, M., Meybodi, M.R., Ebadzadeh, M.M.: Adaptive cooperative particle swarm optimizer. Appl. Intell. **39**, 397–420 (2013)
20. Hashemi, A.B., Meybodi, M.R.: Adaptive parameter selection scheme for PSO: a learning automata approach. In: Proceedings of the 14th International CSI Computer Conference (CSICC'09), IEEE, pp. 403–411 (2009)
21. Hashemi, A.B., Meybodi, M.R.: A note on the learning automata based algorithms for adaptive parameter selection in PSO. Appl. Soft Comput. **11**, 689–705 (2011)
22. Ismail, A., Engelbrecht, A.P.: Self-adaptive particle swarm optimization. In: Bui, L.T., et al. (eds.) SEAL 2012. Lecture Notes in Computer Science, vol. 7673, pp. 228–237 (2012)

23. Ismail, A., Engelbrecht, A.P.: The self-adaptive comprehensive learning particle swarm optimizer. In: Dorigo, M., et al. (eds.) ANTS 2012. Lecture Notes in Computer Science, vol. 7461, pp. 156–167 (2012)
24. Jaillet, P.: Probabilistic traveling salesman problems. Ph.D. thesis, MIT, Cambridge (1985)
25. Jaillet, P.: A priori solution of a traveling salesman problem in which a random subset of the customers are visited. Oper. Res. **36**(6), 929–936 (1988)
26. Jiang, Y., Li, X., Huang, C.: Automatic calibration a hydrological model using a masterslave swarms shuffling evolution algorithm based on self-adaptive particle swarm optimization. Expert Syst. Appl. **40**, 752–757 (2013)
27. Jiang, B., Wang, N., Wang, L.: Particle swarm optimization with age-group topology for multimodal functions and data clustering. Commun. Nonlinear Sci. Numer. Simul. **18**, 3134–3145 (2013)
28. Juang, Y.T., Tung, S.L., Chiu, H.C.: Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. Inform. Sci. **181**, 4539–4549 (2011)
29. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
30. Leu, M.S., Yeh, M.F.: Grey particle swarm optimization. Appl. Soft Comput. **12**, 2985–2996 (2012)
31. Leu, M.S., Yeh, M.F., Wang, S.C.: Particle swarm optimization with grey evolutionary analysis. Appl. Soft Comput. **12**, 4047–4062 (2012)
32. Li, X.L., Li, L.H., Zhang, B.L., Guo, Q.J.: Hybrid self-adaptive learning based particle swarm optimization and support vector regression model for grade estimation. Neurocomputing **118**, 179–190 (2013)
33. Lichtblau, D.: Discrete optimization using Mathematica. In: Callaos, N., Ebisuzaki, T., Starr, B., Abe, J.M., Lichtblau, D. (eds.) World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002). International Institute of Informatics and Systemics, vol. 16, pp. 169–174 (2002)
34. Liu, Y.-H.: A hybrid scatter search for the probabilistic traveling salesman problem. Comput. Oper. Res. **34**(10), 2949–2963 (2007)
35. Liu, Y.-H.: Different initial solution generators in genetic algorithms for solving the probabilistic traveling salesman problem. Appl. Math. Comput. **216**, 125–137 (2010)
36. Lu, H. Chen, W.: Self-adaptive velocity particle swarm optimization for solving constrained optimization problems. J. Glob. Optim. **41**, 427–445 (2008)
37. Marinakis, Y., Marinaki, M.: A hybrid honey bees mating optimization algorithm for the probabilistic traveling salesman problem. In: IEEE Congress on Evolutionary Computation (CEC 2009), 18–21 May, Trondheim (2009)
38. Marinakis, Y., Marinaki, M.: A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. Comput. Oper. Res. **37**, 432–442 (2010)
39. Marinakis, Y., Marinaki, M.: Particle swarm optimization with expanding neighborhood topology for the permutation flowshop scheduling problem. Soft Comput. **17**(7), 1159–1173 (2013)
40. Marinakis, Y., Migdalas, A., Pardalos, P.M.: Expanding neighborhood search GRASP for the probabilistic traveling salesman problem. Optim. Lett. **2**(3), 351–361 (2008)
41. Marinakis, Y., Marinaki, M., Dounias, G.: A hybrid particle swarm optimization algorithm for the vehicle routing problem. Eng. Appl. Artif. Intell. **23**, 463–472 (2010)
42. Modares, H., Alfi, A., Naghibi Sistani, M.B.: Parameter estimation of bilinear systems based on an adaptive particle swarm optimization. Eng. Appl. Artif. Intell. **23**, 1105–1111 (2010)
43. Neshat, M.: FAIPSO: fuzzy adaptive informed particle swarm optimization. Neural Comput. Appl. **23**(1 Suppl.), 95–116 (2013)
44. Nickabadi, A., Ebadzadeh, M.M., Safabakhsh, R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Appl. Soft Comput. **11**, 3658–3670 (2011)
45. Niknam, T.: A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. Appl. Energy **87**, 327–339 (2010)

46. Niknam, T., Farsani, E.A.: A hybrid self-adaptive particle swarm optimization and modified shuffled frog leaping algorithm for distribution feeder reconfiguration. Eng. Appl. Artif. Intell. **23**, 1340–1349 (2010)
47. Niknam, T., Doagou Mojarrad, H., Nayeripour, M.: A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch. Energy **35**, 1764–1778 (2010)
48. Onwubolu, G.C.: TRIBES application to the flow shop scheduling problem. In: Onwubolu, G.C., et al. (eds.) New Optimization Techniques in Engineering, pp. 517–536. Springer, Berlin/Heidelberg (2004)
49. Powell, W.B., Jaillet, P., Odoni, A.: Stochastic and dynamic networks and routing. In: Ball M.O., Magnanti T.L., Momma C.L., Nemhauser G.L. (eds.) Network Routing, Handbooks in Operations Research and Management Science, vol. 8, pp. 141–295. Elsevier Science B. V., Amsterdam (1995)
50. Ratnaweera, A., Halgamuge, S.K., Watson, H.C.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans. Evol. Comput. **8**(3), 240–255 (2004)
51. Senthil Arumugam, M., Rao, M.V.C.: On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. Appl. Soft Comput. **8**, 324–336 (2008)
52. Song, Y.D., Zhang, L., Han, P.: An adaptive tribe-particle swarm optimization. In: Tan, Y., et al. (eds.) ICSI 2011, Part I. Lecture Notes in Computer Science, vol. 6728, pp. 86–92 (2011)
53. Tripathi, P.K., Bandyopadhyay, S., Pal, S.K.: Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Inf. Sci. **177**, 5033–5049 (2007)
54. Wang, J.: Particle swarm optimization with adaptive parameter control and opposition. J. Comput. Inf. Syst. **7**(12), 4463–4470 (2011)
55. Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., Tian, Q.: Self-adaptive learning based particle swarm optimization. Inf. Sci. **181**, 4515–4538 (2011)
56. Wang, J., Lu, H., Dong, Y., Chi, D.: The model of chaotic sequences based on adaptive particle swarm optimization arithmetic combined with seasonal term. Appl. Math. Model. **36**, 1184–1196 (2012)
57. Wang, Y., Zhou, J., Zhou, C., Wang, Y., Qin, H., Lu, Y.: An improved self-adaptive PSO technique for short-term hydrothermal scheduling. Expert Syst. Appl. **39**, 2288–2295 (2012)
58. Xu, G.: An adaptive parameter tuning of particle swarm optimization algorithm. Appl. Math. Comput. **219**, 4560–4569 (2013)
59. Zhan, Z.H., Zhang, J., Li, Y., Chung, H.S.H.: Adaptive particle swarm optimization. IEEE Trans. Syst. Man Cybern. B Cybern. **39**(6), 1362–1381 (2009)
60. Zhang, J., Ding, X.: A multi-swarm self-adaptive and cooperative particle swarm optimization. Eng. Appl. Artif. Intell. **24**, 958–967 (2011)
61. Zhang, W., Liu, Y.: Adaptive particle swarm optimization for reactive power and voltage control in power systems. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. Lecture Notes in Computer Science, vol. 3612, pp. 449–452 (2005)
62. Zhang, L., Mei, T., Liu, Y., Tao, D., Zhou, H.Q.: Visual search reranking via adaptive particle swarm optimization. Pattern Recogn. **44**, 1811–1820 (2011)

Eigendecomposition of the Mean-Variance Portfolio Optimization Model

Fred Mayambala, Elina Rönnberg, and Torbjörn Larsson

Abstract We provide new insights into the mean-variance portfolio optimization problem, based on performing eigendecomposition of the covariance matrix. The result of this decomposition can be given an interpretation in terms of uncorrelated eigenportfolios. When only some of the eigenvalues and eigenvectors are used, the resulting mean-variance problem is an approximation of the original one. A solution to the approximation yields lower and upper bounds on the original mean-variance problem; these bounds are tight if sufficiently many eigenvalues and eigenvectors are used in the approximation. Even tighter bounds are obtained through the use of a linearized error term of the unused eigenvalues and eigenvectors.

We provide theoretical results for the upper bounding quality of the approximate problem and the cardinality of the portfolio obtained, and also numerical illustrations of these results. Finally, we propose an ad hoc linear transformation of the mean-variance problem, which in practice significantly strengthens the bounds obtained from the approximate mean-variance problem.

Keywords Modern portfolio theory • Markowitz model • Eigendecomposition • Quadratic programming

1 Introduction

The mean-variance portfolio optimization model introduced by Markowitz [19] in 1952 continues to be the backbone of modern portfolio theory up to date. This is demonstrated by the huge amount of research that still goes on in this area 63 years later; see for example [24] and [14] for surveys of the field.

F. Mayambala
Makerere University, 7062 Kampala, Uganda
e-mail: fmayambala@cns.mak.ac.ug

E. Rönnberg • T. Larsson (✉)
Linköping University, 58183 Linköping, Sweden
e-mail: elina.ronnberg@liu.se; torbjorn.larsson@liu.se

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_11

The two central inputs to the mean-variance model are the asset returns and the covariance matrix of the returns. However, because the mean-variance model is very sensitive to input data [5, 7], the need for proper methods to estimate the expected returns and the covariance matrix is inevitable. The work by Sharpe [23] is one of the first attempts to formulate models to estimate the covariance matrix. Sharpe developed a single-factor model in which the asset returns depend on only one observable market factor, the market return. This work was later expounded by Lintner [18] and Mossin [21], among others.

However, a single-factor model was later deemed inappropriate for the estimation of asset returns and covariance matrices. This led to the birth of multi-factor models, which make use of a number of observable market variables. For example, Fama and French [10] used a three-factor model to estimate asset returns and the covariance matrix. These factors are the returns on a market portfolio, portfolio size and the book-to-market equity. One more factor, known as the momentum factor of a stock, was added in [6] to give a four-factor model. More studies on multi-factor models have been done in for example [9, 17, 20] and [13]. Attempts to approximate the number of factors required in multi-factor models have been given in [2, 12] and [8], among others. Other methods to estimate covariance matrices have still been sought. For example, in [16] the covariance matrix is estimated as an optimal combination of the sample covariance matrix and the covariance matrix obtained from the single factor model, a method which in the literature is commonly referred to as shrinkage.

In line with the mean-variance model, Fan et al. [11] use a factor model to estimate a covariance matrix with a high dimension compared to the sample size. The obtained covariance estimator is used to demonstrate that under some conditions, it is better than the sample covariance in mean-variance portfolio allocation.

When the set of factors used to estimate the covariance matrix are nonmarket variables, or rather unobservable, then the models are called latent factor models [4]. One latent factor approach for estimating the covariance matrix is principal component analysis. This method involves eigendecomposition of the sample covariance matrix or correlation matrix into eigenvalues and eigenvectors, which are referred to as the principal components. These act as the risk factors. Techniques and theoretical properties for covariance matrices generated using principal component analysis are covered in, for example, [2, 3] and [25].

In relation to portfolio optimization, Avellaneda and Lee [1] use principal component analysis on a sample correlation matrix to generate risk factors, which are shown to carry economic sense. It is shown that the eigenvectors correspond to a new set of portfolios which are uncorrelated, a fact which we exploit in the context of the mean-variance optimization problem with general investment constraints. In essence, principal component analysis is a variable reduction procedure in which most of the variance in the variables is captured within a few largest eigenvalues and their corresponding eigenvectors. The simple question which arises is "What are the implications on the mean-variance model if not all the eigenvalues and corresponding eigenvectors are considered?" This is the question that we address.

We provide insights into the mean-variance problem, based on performing eigendecomposition of the covariance matrix. When using only a subset of the eigenvalues and eigenvectors, the resulting mean-variance problem is a relaxation of the original mean-variance problem. A solution to the relaxed problem provides lower and upper bounds to the original problem; the upper bounding quality is determined by the largest eigenvalue and corresponding eigenvector that is not included in the approximate problem, and the bounds become tight if enough largest eigenvalues and corresponding eigenvectors are used. The addition of an approximate, linearized, error term for the unused eigenvalues and eigenvectors yields even tighter lower and upper bounds on the original mean-variance problem. We further note that the linearized relaxed problem has a solution whose cardinality is governed by the number of eigenvalues and eigenvectors used. The linearized relaxed problem can thus be used to get lower and upper bounds to the mean-variance problem with a cardinality constraint. We also propose an ad hoc linear transformation of the mean-variance model which alters the eigenvalue distribution of the covariance matrix and can be used to greatly improve the lower and upper bounds. The results we provide give a new insight into the problem and create a basis for further research into possible new solution methods.

The remainder of the chapter is organized as follows. Section 2 introduces the approximation method based on eigendecomposition. Inclusion of a linearized error term for the unused eigenvalues and eigenvectors is done in Sect. 3. A numerical illustration of the developed results is provided in Sect. 4, on three different sets of data. As a way to improve the bounds derived in Sect. 2, a transformation of the mean-variance model is made in Sect. 5. Numerical results to demonstrate the effectiveness of the transformation are also given.

2 Eigendecomposition of the Mean-Variance Model

We consider the Mean-Variance (MV) model with n assets of the form

$$\begin{aligned} V^* = \quad & \min_{\mathbf{x}\in\Re^n} \quad \mathbf{x}^{\mathrm{T}}\Sigma\mathbf{x} \\ & \text{s.t.} \quad \mu^{\mathrm{T}}\mathbf{x} \geq \mu_P \\ & \qquad\; \mathbf{x} \in \mathcal{S}, \end{aligned} \tag{MV}$$

where $\mu = (\mu_1, \mu_2, \ldots, \mu_n)^{\mathrm{T}}$ is a vector of expected returns, Σ is a positive semi-definite ($\succeq 0$) covariance matrix of the returns, μ_P is the minimum expected return on the portfolio and $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)^{\mathrm{T}}$ is the vector of fractions of the capital invested in the n assets. Further, $\mathcal{S} \subseteq \Re^n$ is a convex set that describes the possible investment options. It is assumed that the set $\mathcal{S}$ is described by differentiable constraints and that the feasible set of (MV) is nonempty and satisfies Slater's constraint qualification. Note that from the definition of the vector $\mathbf{x}$, it follows that $\mathcal{S} \subseteq \{\mathbf{x} \in \Re^n \mid \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1\}$, where $\mathbf{e}$ is a vector of ones.

Let $\lambda_i,\ i = 1,\ldots,n$, be the eigenvalues of the covariance matrix Σ. Without loss of generality it is assumed that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Further, let $\mathbf{P}$ be an orthonormal matrix whose columns are the eigenvectors of Σ. As is well known, the matrix Σ can then be decomposed into

$$\Sigma = \mathbf{P}\Lambda\mathbf{P}^{\mathrm{T}}, \tag{1}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. We study model (MV) when the covariance matrix Σ has been decomposed as in (1) and analyse the effects of using only a subset of the eigenvalues and eigenvectors.

2.1 Approximation of the Mean-Variance Model

Relationship (1) gives that the covariance matrix Σ can be expressed as

$$\Sigma = \sum_{i=1}^{n} \lambda_i P_i P_i^{\mathrm{T}}, \tag{2}$$

where P_i is the ith eigenvector. Using (2), the objective in (MV) can be expressed as

$$\mathbf{x}^{\mathrm{T}}\Sigma\mathbf{x} = \sum_{i=1}^{n} \lambda_i \mathbf{x}^{\mathrm{T}} P_i P_i^{\mathrm{T}} \mathbf{x} = \sum_{i=1}^{n} \lambda_i (P_i^{\mathrm{T}}\mathbf{x})^2.$$

Introducing the rewritten $z_i = P_i^{\mathrm{T}}\mathbf{x}$, $i = 1, 2, \ldots, n$, the model (MV) can be rewritten as

$$\begin{aligned} V^* = \quad & \min \quad && \sum_{i=1}^{n} \lambda_i z_i^2 \\ & \text{s.t.} && z_i = P_i^{\mathrm{T}}\mathbf{x}, \qquad i = 1,\ldots,n \\ & && \mu^{\mathrm{T}}\mathbf{x} \geq \mu_P \\ & && \mathbf{x} \in \mathcal{S}. \end{aligned} \tag{3}$$

Remark 1. The transformation of the model (MV) into model (3) in essence corresponds to the construction of a mean-variance model in n independent composite assets, or eigenportfolios [1], each of which is a specific portfolio of the n original assets. To see this, we consider for simplicity the easiest case with $\mathcal{S} = \{\mathbf{x} \in \Re^n \mid \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1\}$. Assuming that $\mathbf{e}^{\mathrm{T}}P_i \neq 0$ holds for all i and letting $v_i = (\mathbf{e}^{\mathrm{T}}P_i)z_i,\ i = 1,\ldots,n$, the model (MV) can be rewritten as

$$\begin{aligned} V^* = \quad & \min \quad && \sum_{i=1}^{n} \frac{\lambda_i}{(\mathbf{e}^{\mathrm{T}} P_i)^2} v_i^2 \\ & \text{s.t.} \quad && \sum_{i=1}^{n} \frac{\mu^{\mathrm{T}} P_i}{\mathbf{e}^{\mathrm{T}} P_i} v_i \geq \mu_P \\ & && \sum_{i=1}^{n} v_i = 1. \end{aligned}$$

The interpretation of this rewriting is that the composition of the i^{th} eigenportfolio in terms of fractions of the original assets is $P_i/(\mathbf{e}^{\mathrm{T}} P_i)$, where a negative fraction means short selling, the expected return of the i^{th} eigenportfolio is $(\mu^{\mathrm{T}} P_i)/(\mathbf{e}^{\mathrm{T}} P_i)$ and its variance is $\lambda_i/(\mathbf{e}^{\mathrm{T}} P_i)^2$, while all eigenportfolio covariances are zero. ■

Introducing the index set $I \subseteq \{1, 2, \ldots, n\}$, the model (MV) is approximated by

$$\begin{aligned} V_I^* = \quad & \min \quad && \sum_{i \in I} \lambda_i z_i^2 \\ & \text{s.t.} \quad && z_i = P_i^{\mathrm{T}} \mathbf{x}, \qquad i \in I \\ & && \mu^{\mathrm{T}} \mathbf{x} \geq \mu_P \\ & && \mathbf{x} \in \mathcal{S}. \end{aligned} \tag{4}$$

Note that problem (4) is always feasible, since (MV) is feasible. The following theorem shows that problem (4) is a relaxation of (3) and (MV), thus providing a lower bound on V^*, and that its solution also provides an upper bound.

Theorem 1. *Let* $\mathbf{x}_I^*$ *be optimal in* (4) *and define* $\hat{V}_I^* = \mathbf{x}_I^{*\mathrm{T}} \Sigma \mathbf{x}_I^*$. *Then*

$$V_I^* \leq V^* \leq \hat{V}_I^*.$$

Proof. Let $\mathbf{x}^*$ be optimal in (3). We note that

$$\begin{aligned} V^* &= \sum_{i=1}^{n} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}^*)^2 = \sum_{i \in I} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}^*)^2 + \sum_{i \notin I} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}^*)^2 \\ &\geq \sum_{i \in I} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}^*)^2 \geq \sum_{i \in I} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}_I^*)^2 = V_I^*, \end{aligned}$$

where the first inequality holds because $\Sigma \succeq 0$ and the second because $\mathbf{x}_I^*$ is optimal in (4), which completes the proof for the first inequality in the theorem. For the second inequality,

$$V^* = \sum_{i=1}^{n} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}^*)^2 \leq \sum_{i=1}^{n} \lambda_i (P_i^{\mathrm{T}} \mathbf{x}_I^*)^2 = \hat{V}_I^*$$

where the inequality holds because $\mathbf{x}^*$ is optimal in (3). □

Note that if $\lambda_i > 0$ holds for all $i \in I$, then the objective function of (4) is strictly convex with respect to the variables $z_i, i \in I$, which together with the convexity of the problem imply that these variables have unique optimal values. The lower bound V_I^* is clearly nondecreasing if the set I is augmented with some $i \in \{1,\ldots,n\} \setminus I$.

We next analyse the behaviour of the upper bound $\hat{V}_I^*$ with respect to the set I. Let us define

$$\Sigma_I = \sum_{i \in I} \lambda_i P_i P_i^{\mathrm{T}}.$$

The next theorem gives a bound on the deviation of the upper bound $\hat{V}_I^*$ from V^*.

Theorem 2. *Let λ_{max} be the largest eigenvalue for $\Sigma - \Sigma_I$ and let $\varepsilon > 0$. If I is chosen so that $\lambda_{max} \|\mathbf{x}_I^*\|_2^2 / V_I^* \leq \varepsilon$, then*

$$\frac{\hat{V}_I^* - V^*}{V^*} \leq \varepsilon.$$

Proof. Using that the Rayleigh quotient of a symmetric matrix is bounded from above by the largest eigenvalue, it holds for any $\mathbf{x}$ that

$$\mathbf{x}^{\mathrm{T}} (\Sigma - \Sigma_I) \mathbf{x} \leq \lambda_{max} \|\mathbf{x}\|_2^2.$$

Thus, if $\mathbf{x}_I^*$ is optimal in (4), then

$$\hat{V}_I^* - V_I^* = \mathbf{x}_I^{*\mathrm{T}} \Sigma \mathbf{x}_I^* - \mathbf{x}_I^{*\mathrm{T}} \Sigma_I \mathbf{x}_I^* = \mathbf{x}_I^{*\mathrm{T}} (\Sigma - \Sigma_I) \mathbf{x}_I^* \leq \lambda_{max} \|\mathbf{x}_I^*\|_2^2.$$

Hence,

$$\frac{\hat{V}_I^* - V^*}{V^*} \leq \frac{\hat{V}_I^* - V_I^*}{V_I^*} \leq \frac{\lambda_{max} \|\mathbf{x}_I^*\|_2^2}{V_I^*}$$

where the first inequality follows from Theorem 1. □

Corollary 1. *Assume that $\mathcal{S} \subseteq \Re_+^n$ and let λ_{max} be the largest eigenvalue for $\Sigma - \Sigma_I$. Then*

$$\hat{V}_I^* - V^* \leq \lambda_{max}$$

holds.

Proof. Since $\{\mathbf{x} \in \mathcal{S} \mid \mu^{\mathrm{T}} \mathbf{x} \geq \mu_P\} \subseteq \{\mathbf{x} \in \Re^n \mid \|\mathbf{x}\|_1 = 1\}$, then $\|\mathbf{x}_I^*\|_2^2 \leq \|\mathbf{x}_I^*\|_1^2 = 1$ holds. Therefore,

$$\hat{V}_I^* - V^* \leq \hat{V}_I^* - V_I^* \leq \mathbf{x}_I^{*\mathrm{T}} (\Sigma - \Sigma_I) \mathbf{x}_I^* \leq \lambda_{max} \|\mathbf{x}_I^*\|_2^2 \leq \lambda_{max} \|\mathbf{x}_I^*\|_1^2 = \lambda_{max}.$$

□

Hence, a near-optimal solution to (MV) with an a priori known quality can be found by selecting an appropriate set of eigenvalues and corresponding eigenvectors in (4). The following corollary also follows from Theorem 2.

Corollary 2. *Let* $\mathbf{x}_I^*$ *be optimal in* (4) *and let* $\mathbb{X}_I$ *be the set of all such points. Consider a fixed cardinality of the set I, say k. Then*

$$\min_{I:|I|=k} \max_{\mathbf{x}_I^* \in \mathbb{X}_I} \left(\hat{V}_I^* - V^*\right)$$

is achieved for the choice $I = \{1, 2, \ldots, k\}$.

We conclude that, for a fixed cardinality of the set I, the best possible worst-case outcome of the gap $\hat{V}_I^* - V^*$ is obtained by constructing the approximate problem (4) from $|I|$ largest eigenvalues and corresponding eigenvectors.

3 An Improved Approximation Strategy

We have shown in Sect. 2 that the approximation of (MV) becomes better as we increase the number of eigenvalues and eigenvectors according to decreasing order of eigenvalues. As a way to further improve the approximation and get tighter bounds, we can add a linear approximation of the error introduced by not making use of all eigenvalues and eigenvectors in (4). The introduction of a linearized error term was inspired by the work in [15].

3.1 A Linearized Error Term

The objective function in (MV) can be rewritten as

$$\mathbf{x}^{\mathrm{T}}\Sigma\mathbf{x} = \mathbf{x}^{\mathrm{T}}\Sigma_I\mathbf{x} + \mathbf{x}^{\mathrm{T}}\left(\Sigma - \Sigma_I\right)\mathbf{x}.$$

The first term is the objective of (4) and we thus define the function $e : \Re^n \to \Re$ with

$$e(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}(\Sigma - \Sigma_I)\mathbf{x}$$

as the error introduced by using $|I|$ eigenvalues and eigenvectors of the total n. This function is convex, since the matrix $\Sigma - \Sigma_I$ has $n - |I|$ eigenvalues that are λ_i, $i \notin I$, and the remaining are zero.

A Taylor series expansion of e at some point $\bar{\mathbf{x}} \in \Re^n$ up to the linear term gives the function $\hat{e} : \Re^n \to \Re$ defined by

$$\hat{e}(\mathbf{x}) = \bar{\mathbf{x}}^{\mathrm{T}}(\Sigma - \Sigma_I)\bar{\mathbf{x}} + 2\left[(\Sigma - \Sigma_I)\bar{\mathbf{x}}\right]^{\mathrm{T}}(\mathbf{x} - \bar{\mathbf{x}}).$$

However noting that the function $\hat{e}$ can take negative values although e cannot, we choose the approximate error function as $\bar{e} : \Re^n \to \Re_+$ defined by

$$\bar{e}(\mathbf{x}) = \max\{\hat{e}(\mathbf{x}), 0\}. \tag{5}$$

Model (4) with error term (5) becomes

$$\begin{aligned} M_I^* = \quad & \min \quad && \sum_{i\in I} \lambda_i z_i^2 + \bar{e}(\mathbf{x}) \\ & \text{s.t.} \quad && z_i = P_i^{\mathrm{T}}\mathbf{x}, \qquad i \in I && \text{(6a)} \\ & && \mu^{\mathrm{T}}\mathbf{x} \geq \mu_P && \text{(6b)} \\ & && \mathbf{x} \in \mathcal{S}. && \text{(6c)} \end{aligned}$$

By introducing an auxiliary variable w, model (6) can be rewritten as

$$\begin{aligned} M_I^* = \quad & \min \quad && \sum_{i\in I} \lambda_i z_i^2 + w \\ & \text{s.t.} \quad && w \geq 0 \\ & && w \geq \hat{e}(\mathbf{x}) \\ & && \text{(6a)},\ \text{(6b)},\ \text{(6c)}. \end{aligned} \tag{7}$$

The following theorem shows that (6) is also a relaxation of (MV) and that it also provides upper bounds.

Theorem 3. *Let* $\mathbf{x}_E^*$ *be optimal in* (6) *and define* $\hat{M}_I^* = \mathbf{x}_E^{*\mathrm{T}}\Sigma\mathbf{x}_E^*$. *Then*

$$M_I^* \leq V^* \leq \hat{M}_I^*.$$

Proof. Using the definition of e, its convexity and nonnegativity, and the optimality of $\mathbf{x}_E^*$ in (6), we obtain

$$\begin{aligned} V^* &= \sum_{i=1}^{n} \lambda_i (P_i^{\mathrm{T}}\mathbf{x}^*)^2 = \sum_{i\in I} \lambda_i (P_i^{\mathrm{T}}\mathbf{x}^*)^2 + \sum_{i\notin I} \lambda_i (P_i^{\mathrm{T}}\mathbf{x}^*)^2 \\ &= \sum_{i\in I} \lambda_i (P_i^{\mathrm{T}}\mathbf{x}^*)^2 + e(\mathbf{x}^*) \geq \sum_{i\in I} \lambda_i (P_i^{\mathrm{T}}\mathbf{x}^*)^2 + \bar{e}(\mathbf{x}^*) \\ &\geq \sum_{i\in I} \lambda_i (P_i^{\mathrm{T}}\mathbf{x}_E^*)^2 + \bar{e}(\mathbf{x}_E^*) = M_I^*. \end{aligned}$$

This proves the first inequality in the theorem. The proof for the second inequality is as in the proof of Theorem 1. □

The following proposition shows that (6) gives a lower bound that is at least as good as that of (4) for the same set I.

Proposition 1. $M_I^* \geq V_I^*$

Proof. Let $\mathbf{x}_E^*$ and $\mathbf{x}_I^*$ be optimal in (6) and (4), respectively. Using that $\bar{e}(\mathbf{x}) \geq 0$, we then see that

$$M_I^* = \mathbf{x}_E^{*\mathrm{T}} \Sigma_I \mathbf{x}_E^* + \bar{e}(\mathbf{x}_E^*) \geq \mathbf{x}_E^{*\mathrm{T}} \Sigma_I \mathbf{x}_E^* \geq \mathbf{x}_I^{*\mathrm{T}} \Sigma_I \mathbf{x}_I^* = V_I^*.$$

□

Then a result analogous to Theorem 2 is given.

Theorem 4. *Let λ_{max} be the largest eigenvalue of $\Sigma - \Sigma_I$ and let $\varepsilon > 0$. If I is chosen so that $\lambda_{max} \|\mathbf{x}_E^*\|_2^2 / M_I^* \leq \varepsilon$, then*

$$\frac{\hat{M}_I^* - V^*}{V^*} \leq \varepsilon.$$

Proof. Following similar arguments as in the proof of Theorem 2,

$$\hat{M}_I^* - M_I^* = \mathbf{x}_E^{*\mathrm{T}} (\Sigma - \Sigma_I) \mathbf{x}_E^* - \bar{e}(\mathbf{x}_E^*) \leq \lambda_{max} \|\mathbf{x}_E^*\|_2^2 - \bar{e}(\mathbf{x}_E^*) \leq \lambda_{max} \|\mathbf{x}_E^*\|_2^2.$$

Hence,

$$\frac{\hat{M}_I^* - V^*}{V^*} \leq \frac{\hat{M}_I^* - M_I^*}{M_I^*} \leq \frac{\lambda_{max} \|\mathbf{x}_E^*\|_2^2}{M_I^*}.$$

□

Results analogous to Corollaries 1 and 2 also follow easily from Theorem 4.

Corollary 3. *Assume that $\mathcal{S} \subseteq \Re_+^n$ and let λ_{max} be the largest eigenvalue for $\Sigma - \Sigma_I$. Then*

$$\hat{M}_I^* - V^* \leq \lambda_{max}$$

holds.

Proof. Following similar arguments as in the proof of Corollary 1, we see that

$$\begin{aligned} \hat{M}_I^* - V^* \leq \hat{M}_I^* - M_I^* &= \mathbf{x}_E^{*\mathrm{T}} \Sigma \mathbf{x}_E^* - \mathbf{x}_E^{*\mathrm{T}} \Sigma_I \mathbf{x}_E^* - \bar{e}(\mathbf{x}_E^*) \\ &\leq \mathbf{x}_E^{*\mathrm{T}} (\Sigma - \Sigma_I) \mathbf{x}_E^* \leq \lambda_{max} \|\mathbf{x}_E^*\|_2^2 \leq \lambda_{max} \|\mathbf{x}_E^*\|_1^2 = \lambda_{max}. \end{aligned}$$

□

Corollary 4. *Let $\mathbf{x}_E^*$ be optimal in (6) and let $\mathbb{X}_E$ be the set of all such points. Consider a fixed cardinality of the set I, say k. Then*

$$\min_{I:|I|=k} \max_{\mathbf{x}_E^* \in \mathbb{X}_E} \left(\hat{M}_E^* - V^* \right)$$

is achieved for the choice $I = \{1, 2, \ldots, k\}$.

We next show that if the approximate problem (6) is constructed from an optimal solution to (MV), then in fact it becomes equivalent to (MV) with respect to the optimal objective value.

Theorem 5. *If the linearization point* $\bar{\mathbf{x}}$ *in* (6) *is chosen as optimal in* (MV)*, then* $\bar{\mathbf{x}}$ *is also optimal in* (6) *and* $M_I^* = V^*$ *holds.*

Proof. Problem (6) can be rewritten as

$$\begin{array}{ll} \min & \mathbf{x}^{\mathrm{T}}\Sigma_I\mathbf{x} + \max\{\hat{e}(\mathbf{x}), 0\} \\ \text{s.t.} & \text{(6b)},\ \text{(6c)}. \end{array} \tag{8}$$

The two problems (MV) and (8) have the same feasible set and they are both convex. Let us consider first the problem

$$\begin{array}{ll} \min & \mathbf{x}^{\mathrm{T}}\Sigma_I\mathbf{x} + \hat{e}(\mathbf{x}) \\ \text{s.t.} & \text{(6b)},\ \text{(6c)}. \end{array} \tag{9}$$

Then, from

$$\nabla\left(\hat{e}(\mathbf{x})\right)_{\mathbf{x}=\bar{\mathbf{x}}} = \nabla\left(\mathbf{x}^{\mathrm{T}}(\Sigma - \Sigma_I)\mathbf{x}\right)_{\mathbf{x}=\bar{\mathbf{x}}}$$

it follows that

$$\nabla\left(\mathbf{x}^{\mathrm{T}}\Sigma_I\mathbf{x} + \hat{e}(\mathbf{x})\right)_{\mathbf{x}=\bar{\mathbf{x}}} = \nabla\left(\mathbf{x}^{\mathrm{T}}\Sigma\mathbf{x}\right)_{\mathbf{x}=\bar{\mathbf{x}}}.$$

This implies that any KKT point for (MV) is also a KKT point for (9), from which we conclude that $\bar{\mathbf{x}}$ is also an optimal solution in (9), and therefore also optimal in

$$\begin{array}{ll} \min & \mathbf{x}^{\mathrm{T}}\Sigma_I\mathbf{x} + w \\ \text{s.t.} & w \geq \hat{e}(\mathbf{x}) \\ & \text{(6b)},\ \text{(6c)}. \end{array} \tag{10}$$

Then the optimal value of w in (10) is

$$\bar{w} = \hat{e}(\bar{\mathbf{x}}) = \bar{\mathbf{x}}^{\mathrm{T}}(\Sigma - \Sigma_I)\bar{\mathbf{x}}.$$

Noting that $\hat{e}(\bar{\mathbf{x}}) \geq 0$, we conclude that $(\bar{\mathbf{x}}, \bar{w})$ is optimal in

$$\begin{array}{ll} \min & \mathbf{x}^{\mathrm{T}}\Sigma_I\mathbf{x} + w \\ \text{s.t.} & w \geq 0 \\ & w \geq \hat{e}(\mathbf{x}) \\ & \text{(6b)},\ \text{(6c)}, \end{array}$$

which implies that $\bar{\mathbf{x}}$ is optimal in (8) and (6). Further still, if $\bar{\mathbf{x}}$ is optimal in (6), then, since $\hat{e}(\bar{\mathbf{x}}) = \bar{\mathbf{x}}^{\mathrm{T}}(\Sigma - \Sigma_I)\bar{\mathbf{x}} \geq 0$,

$$\begin{aligned} M_I^* =& \bar{\mathbf{x}}^{\mathrm{T}}\Sigma_I\bar{\mathbf{x}} + \max\{\hat{e}(\bar{\mathbf{x}}), 0\} = \bar{\mathbf{x}}^{\mathrm{T}}\Sigma_I\bar{\mathbf{x}} + \hat{e}(\bar{\mathbf{x}}) = \\ =& \bar{\mathbf{x}}^{\mathrm{T}}\Sigma_I\bar{\mathbf{x}} + \bar{\mathbf{x}}^{\mathrm{T}}(\Sigma - \Sigma_I)\bar{\mathbf{x}} = \bar{\mathbf{x}}^{\mathrm{T}}\Sigma\bar{\mathbf{x}} = V^*. \end{aligned}$$

□

3.2 Cardinality of the Solution

We here show that a further analysis of problems (4) and (6) provides an insight into the cardinality of their solutions. Here, Card$(\cdot)$ denotes the cardinality of a vector.

Theorem 6. *Assume that* $\mathcal{S} = \{\mathbf{x} \in \Re^n \mid \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1,\ \mathbf{x} \geq 0\}$. *Let* z_I^* *be optimal in* (4). *Then there exists an* x_I^* *that is optimal in* (4) *with*

$$\mathrm{Card}(\mathbf{x}_I^*) \leq |I| + 2.$$

Proof. For fixed $z_i^*, i \in I$, any $\mathbf{x}$ that is feasible in the system

$$\begin{aligned} & z_i^* = P_i^{\mathrm{T}}\mathbf{x}, \qquad i \in I \\ & \mu^{\mathrm{T}}\mathbf{x} \geq \mu_P \\ & \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1 \\ & \mathbf{x} \geq 0 \end{aligned} \tag{11}$$

is optimal in (4). Any extreme point of the set described by system (11) has at most $|I| + 2$ nonzero components. □

Theorem 7. *Assume that* $\mathcal{S} = \{\mathbf{x} \in \Re^n \mid \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1,\ \mathbf{x} \geq 0\}$. *Let* z_E^* *be optimal in* (6). *Then there exists an* x_E^* *that is optimal in* (6) *with*

$$\mathrm{Card}(\mathbf{x}_E^*) \leq |I| + 3.$$

Proof. If augmenting system (11) with the nonnegative variable w and the constraint $w \geq \hat{e}(\bar{x})$, then any extreme point of the corresponding set has at most $|I| + 3$ nonzero components. Further, $w = 0$ can hold at an extreme point, in which case the vector $\mathbf{x}$ contains at most $|I| + 3$ nonzero components. □

Corollary 5. *Assume that* $\mathcal{S} = \{\mathbf{x} \in \Re^n \mid \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1,\ \mathbf{x} \geq 0\}$. *Then optimal solutions to* (4) *and* (6) *that are extreme points with respect to* $\mathbf{x}$ *and* $(\mathbf{x}, w)$, *respectively, correspond to portfolios that include at most* $|I| + 2$ *and* $|I| + 3$ *assets, respectively.*

This results is of interest since linear programming based solvers can be expected to produce extreme point solutions to (4) and (6), even though alternative solutions exist.

Remark 2. For the case $\mathcal{S} = \{\mathbf{x} \in \Re^n \mid \mathbf{e}^T\mathbf{x} = 1,\ \mathbf{x} \geq 0\}$, an optimal solution of (6) with cardinality K provides both lower and upper bounds to the cardinality constrained mean-variance problem

$$\begin{aligned} \min_{\mathbf{x}\in\Re^n} \quad & \mathbf{x}^T\Sigma\mathbf{x} \\ \text{s.t.} \quad & \mu^T\mathbf{x} \geq \mu_P \\ & \mathbf{e}^T\mathbf{x} = 1 \\ & \text{Card}(\mathbf{x}) \leq K \\ & \mathbf{x} \geq 0. \end{aligned}$$

■

We finally note that an optimal solution obtained from solving (MV) using the state-of-the-art software (like CPLEX, which we used) usually contains many very small variable values, which can make it difficult when it comes to deciding which variables shall actually take the value zero and which shall not. Note that (7) also suffers from this numerical difficulty. To circumvent such a problem, one can solve problem (7) in order to obtain optimal values $z_i^*,\ i \in I$, and then resolve the problem in the variables $\mathbf{x}$ only, that is

$$\begin{aligned} \min \quad & \bar{e}(\mathbf{x}) + \text{constant} \\ \text{s.t.} \quad & z_i^* = P_i^T\mathbf{x}, \qquad i \in I \\ & \mu^T\mathbf{x} \geq \mu_P \\ & \mathbf{x} \in \mathcal{S}, \end{aligned}$$

where constant $= \sum_{i\in I} \lambda_i z_i^{*2}$. In contrast to (MV), this problem is a linear program and a solver will produce an optimal basic feasible solution that contains zero-valued variables.

4 Numerical Illustrations

In this section we perform numerical computations to demonstrate the results given in Sects. 2 and 3 on three data sets.

One of the data sets, with 225 assets, is obtained from the OR-library [22]. The other two data sets, with 500 and 1,000 assets, are obtained using historical data from NYSE. We collected daily opening and closing prices for 500 and 1,000

Table 1 Problem instances used

n	$[\min \mu_i,\ \max \mu_i]$	Problem instance	μ_P	V^*	card($\mathbf{x}^*$)
225	$[-0.008489, 0.003971]$	225A	0.003	5.15395e−4	8
		225B	0.001	3.25288e−4	14
500	$[-0.00034669, 0.0077343]$	500A	0.003	1.71429e−3	10
		500B	0.001	8.03068e−5	31
1,000	$[-0.00051135,\ 0.0076479]$	1000A	0.003	1.30006e−3	13
		1000B	0.001	6.55035e−5	40

randomly selected assets from NYSE from the year 2005 to 2014 and used these to calculate the daily expected returns and the covariance matrices using MATLAB's inbuilt functions. For each of the three data sets, we construct two different problem instances which are used for the computations. One of the instances is a high-risk and high-return portfolio optimization problem and the other a low-risk and low-return. The instances are summarized in Table 1. In all instances, we use the set $\mathcal{S} = \{\mathbf{x} \in \Re^n \mid \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1,\ \mathbf{x} \geq 0\}$.

4.1 Upper and Lower Bounds

The purpose of this section is to illustrate the effect of increasing the number of eigenvalues and eigenvectors, on the quality of an optimal solution obtained by solving (4) and (6) for the same set I. Eigenvalues and eigenvectors are added in order of increasing eigenvalues and the lower and upper bounds are plotted against $|I| = k$ for the six problem instances in Table 1.

From Figs. 1, 2 and 3, it can be seen that the addition of a linearized error term improves both the lower and the upper bounds for the optimal value V^*. For the high-risk portfolios, near-optimal solutions can in fact be obtained by using less than 20 % of the eigenvalues and corresponding eigenvectors. However, low-risk portfolios require a large number of eigenvalues and eigenvectors to attain an optimal solution. As noted from the figures, a good upper bound of the optimal objective value can always be obtained using the approximate problem (6) and approximately 20 % of the largest eigenvalues and their corresponding eigenvectors.

4.2 Deviation in Solution

The purpose of the experimental results presented here is to illustrate the conclusions of Corollaries 1 and 3. For each of the six problem instances in Table 1, we compute the difference between the upper bound obtained by eigendecomposition and the optimal solution of the problem, for an increasing number, k, of eigenvalues

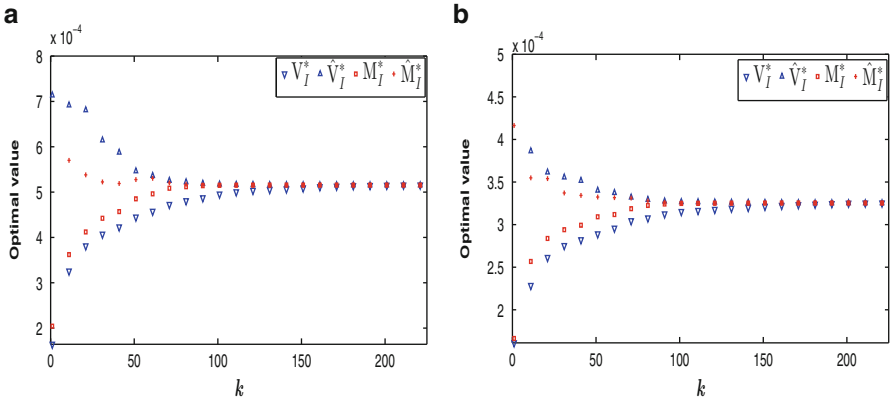


Fig. 1 Lower and upper bounds versus the number of eigenvalues for 225 assets. (**a**) Instance 225A. (**b**) Instance 225B

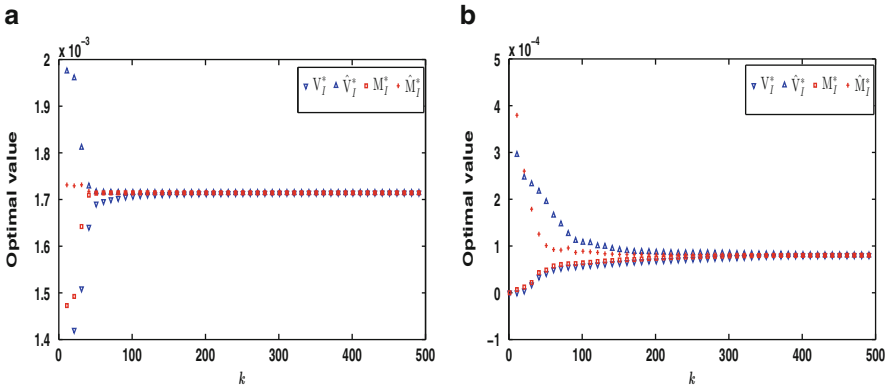


Fig. 2 Lower and upper bounds versus the number of eigenvalues for 500 assets. (**a**) Instance 500A. (**b**) Instance 500B

and corresponding eigenvectors used in problems (4) and (6). The deviation $\hat{V}_I^* - V^*$, for the case without an error term, is compared with the deviation $\hat{M}_I^* - V^*$, for the case with a linearized error term, and also with the theoretical upper bound $\lambda_{\max}$.

For 225 assets in Fig. 4, it is seen that the approximate model that includes a linearized error term requires less number of eigenvalues and eigenvectors to produce a very good approximation of the solution to (MV). Note also that the computed deviations are always significantly smaller than the upper bound $\lambda_{\max}$.

Considering the case of 500 assets in Fig. 5, less than 20 % of the eigenvalues and eigenvectors are required to produce an exact optimal solution for instance 500A. However, the cardinality of the optimal solution is so high for instance 500B that it requires almost all the eigenvalues and eigenvectors to produce an exact solution.

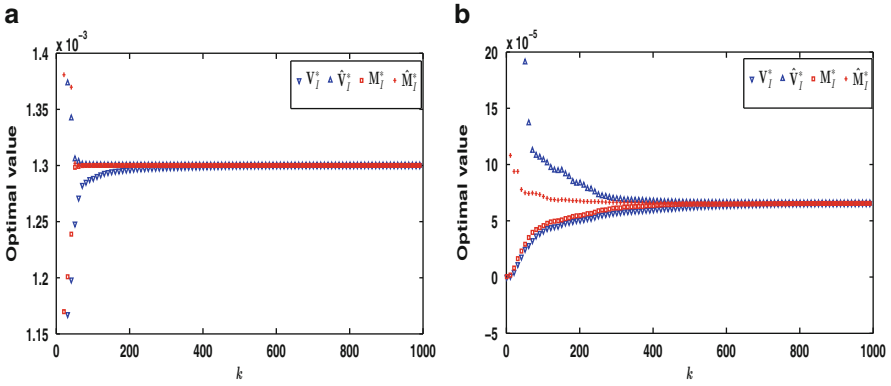


Fig. 3 Lower and upper bounds versus the number of eigenvalues for 1,000 assets. (**a**) Instance 1000A. (**b**) Instance 1000B

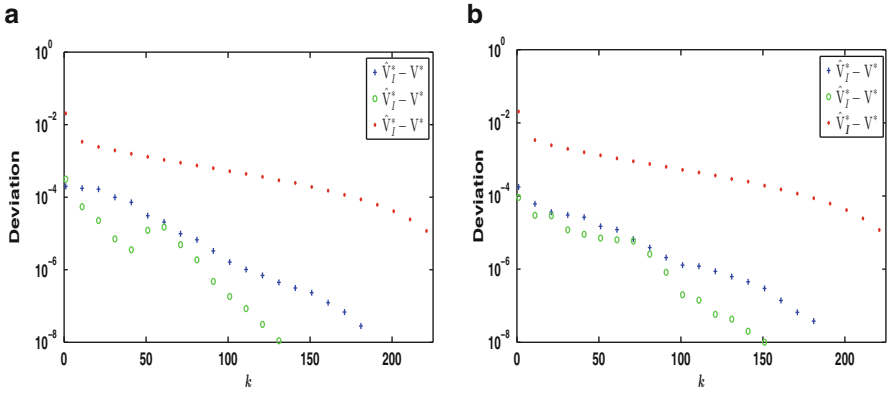


Fig. 4 Upper bounding quality versus the number of eigenvalues for 225 assets. (**a**) Instance 225A. (**b**) Instance 225B

The solution quality is improved greatly on addition of a linearized error term for the case of 1,000 assets, shown in Fig. 6. For the high-return instance 1000A, the solution can be obtained with less than 100 eigenvalues and eigenvectors, while for the low-risk instance 1000B, much more eigenvalues and eigenvectors are needed.

For the instances considered, it can be seen from Figs. 4, 5 and 6 that computation of an exact optimal solution, or a very good approximation, can be obtained with a number of eigenvalues and eigenvectors that is less than n and in some cases significantly less than n. All the computations also show that inclusion of a linearized error term further improves the solution quality and allows a less number of eigenvalues and eigenvectors to yield a better approximation. The instances with a lower value of portfolio return require more eigenvalues and eigenvectors for a good

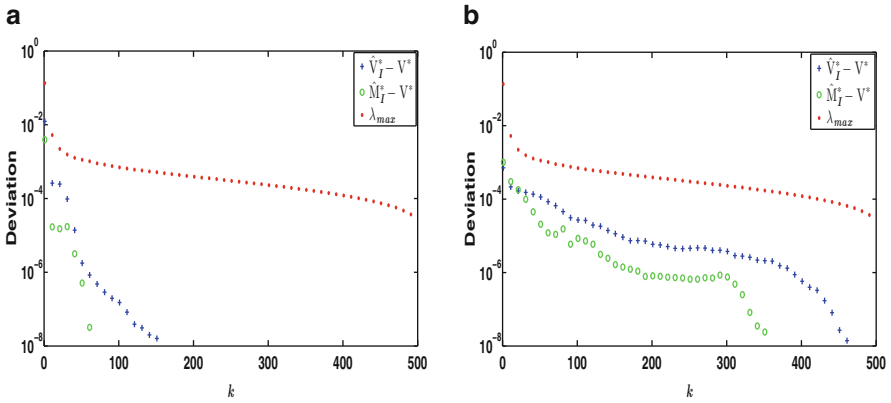


Fig. 5 Upper bounding quality versus the number of eigenvalues for 500 assets. (**a**) Instance 500A. (**b**) Instance 500B

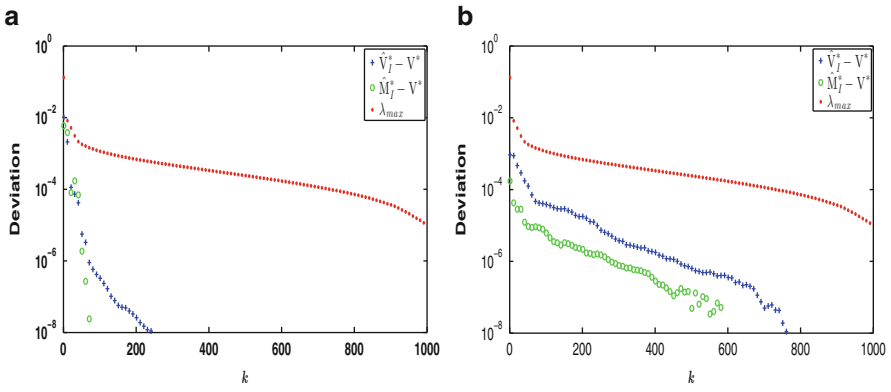


Fig. 6 Upper bounding quality versus the number of eigenvalues for 1,000 assets. (**a**) Instance 1000A. (**b**) Instance 1000B

approximation, as compared to the instances with a higher value of portfolio return. This is attributed to the fact that portfolios with lower portfolio returns contain more assets, since this lessens the risk taken, and hence require a higher number of eigenvalues and eigenvectors to be approximated.

4.3 Efficient Frontier

We compute an approximation of the efficient frontier using both models (4) and (6) for 10, 20, 40 and 50 % of the total number of eigenvalues and eigenvectors. Again, the eigenvalues are included according to decreasing values. The lower bounds V_I^*

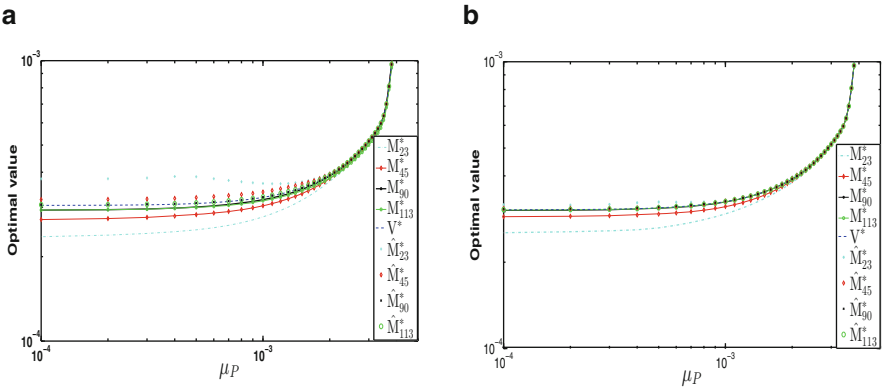


Fig. 7 Exact and approximate efficient frontiers for 225 assets. (**a**) With no error term. (**b**) With a linearized error term

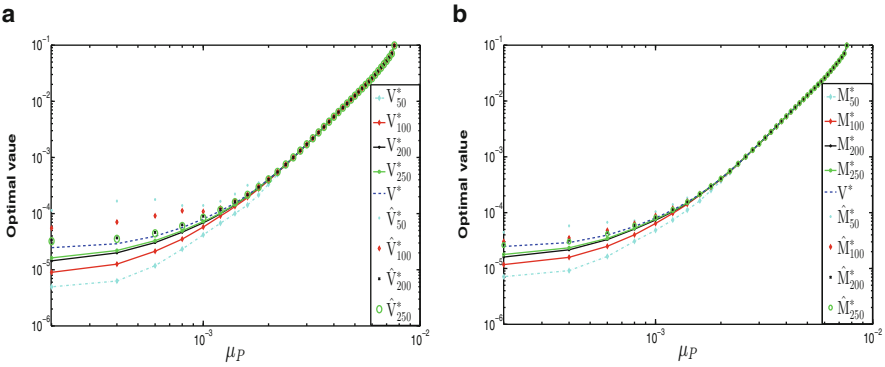


Fig. 8 Exact and approximate efficient frontiers for 500 assets. (**a**) With no error term. (**b**) With a linearized error term

and M_I^* obtained from (4) and (6), respectively, and the corresponding upper bounds $\hat{V}_I^*$ and $\hat{M}_I^*$, are compared with the correct optimal value V^*. The results are shown in Figs. 7, 8 and 9.

For the case of 225 assets, the efficient frontier can be fairly well approximated using 45 largest eigenvalues and corresponding eigenvectors, as shown in Fig. 7. The approximation improves with an increasing portfolio return, and it gets even better when a linearized error term is added. In Fig. 8, showing the case of 500 assets, the approximate efficient frontier also improves with increasing values of portfolio return and number of eigenvalues and eigenvectors. For higher values of portfolio return, 50 eigenvalues and eigenvectors are sufficient whereas 250 eigenvalues and eigenvectors are needed for smaller values of portfolio return. For the case of 1,000 assets, in Fig. 9, the behaviour is similar to those of the other instances.

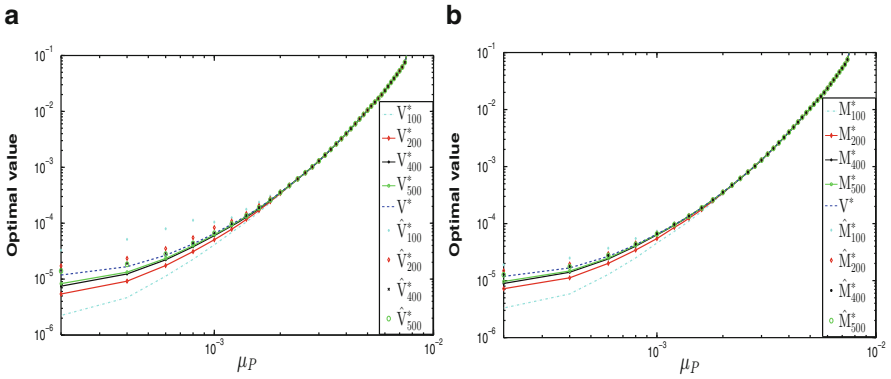


Fig. 9 Exact and approximate efficient frontiers for 1,000 assets. (**a**) With no error term. (**b**) With a linearized error term

As observed from Figs. 7, 8 and 9, the whole efficient frontier can be well approximated using less than 50 % of the total number of eigenvalues and eigenvectors. The amount of computations needed to find these approximate efficient frontiers are actually light compared to solving the model (MV) using the state-of-the-art software. For practitioners who need a quick computation of the frontier, this strategy can thus be handy.

4.4 Cardinality of the Solution

We here study the cardinality of an optimal solution $\mathbf{x}_E^*$ of (6) for the problem instances given in Table 1. For an increasing number of eigenvalues and eigenvectors, again added in decreasing eigenvalue order, we compute the cardinality of each optimal solution, $\text{Card}(\mathbf{x}_E^*)$, for $|I| = k$ eigenvalues and corresponding eigenvectors. Here, any value of the optimal solution $\mathbf{x}_E^*$ smaller than 10^{-5} is considered to be zero.

For problem instance 225A in Fig. 10a, less than 50 eigenvalues and eigenvectors are required to attain the correct cardinality of the optimal solution. However, more than 50 eigenvalues and eigenvectors are required for problem instance 225B.

In Fig. 11a, a high number of eigenvalues and eigenvectors is required to reach the correct cardinality, whereas a much smaller number is required for instance 500A in Fig. 11b.

As shown in Fig. 12, the correct cardinality of the optimal solution is obtained with much fewer eigenvalues for problem instance 1000A than for instance 1000B.

Note from Figs. 10, 11 and 12 that the cardinality of the optimal solution $\mathbf{x}_E^*$ never exceeds the correct cardinality of the optimal solution of (MV), which is given in Table 1. Common for all three data sets is that the low-risk instances require more eigenvalues and eigenvectors in problem (6) in order to reach the correct cardinality

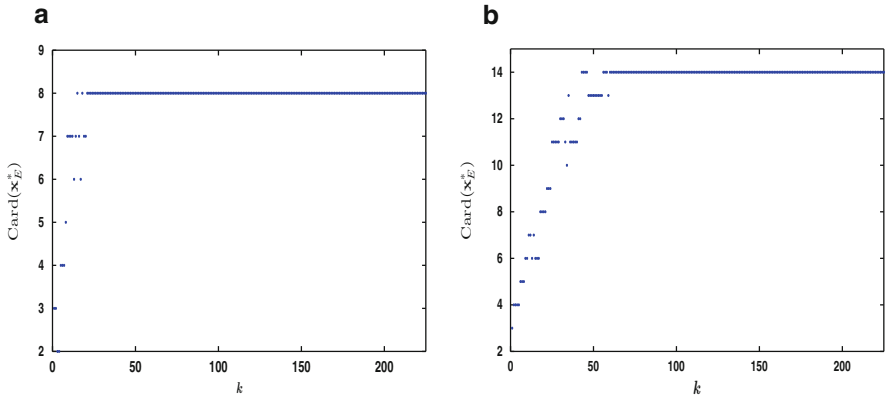


Fig. 10 Cardinality of an optimal solution obtained from (6) versus the number of eigenvalues and eigenvectors considered for the 225 asset instances. (**a**) Instance 225A. (**b**) Instance 225B

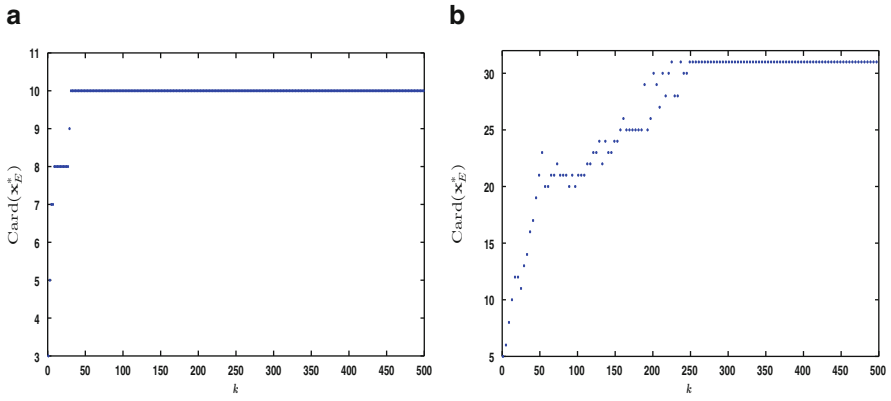


Fig. 11 Cardinality of an optimal solution obtained from (6) versus the number of eigenvalues and eigenvectors considered for the 500 asset instances. (**a**) Instance 500A. (**b**) Instance 500B

of an optimal solution. This is because these instances have optimal solutions that include more assets, which in turn, according to the results in Sect. 3.2, require more eigenvalues and eigenvectors to be considered.

5 A Proposed Transformation

According to Theorems 2 and 4, the quality of an optimal solution obtained from solving problems (4) or (6) depends on the eigenvalue distribution of the covariance matrix. It is then natural to try to improve this quality by changing the eigenvalue distribution through a linear transformation. However, in order to avoid that such

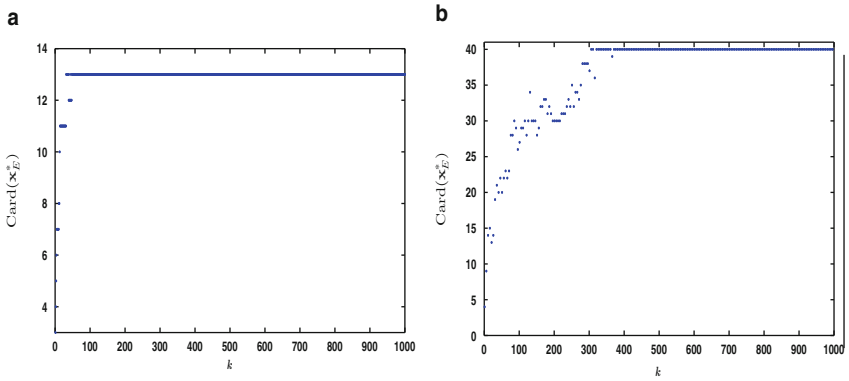


Fig. 12 Cardinality of an optimal solution obtained from (6) versus the number of eigenvectors and eigenvalues considered for the 1,000 asset instances. (**a**) Instance 1000A. (**b**) Instance1000B

a transformation turns the constraints $\mathbf{x} \geq 0$ into general linear constraints, only diagonal transformations are considered. We here propose the transformation

$$y_i = (\mu_i - r_b)x_i, \qquad i = 1, 2, \ldots, n \tag{12}$$

for a constant parameter r_b. In order to have a one-to-one correspondence between the variables x_i and y_i, we insert the requirement that $r_b \neq \mu_i, \quad i = 1, \ldots, n$. The objective in (MV) then becomes

$$\mathbf{x}^T \Sigma \mathbf{x} = \mathbf{y}^T \mathbf{Q} \mathbf{y},$$

where

$$\mathbf{Q} = \mathbf{D}\Sigma\mathbf{D},$$

with $\mathbf{D} = \text{diag}\left(\frac{1}{\mu_1 - r_b}, \frac{1}{\mu_2 - r_b}, \ldots, \frac{1}{\mu_n - r_b}\right)$. We note that the matrix $\mathbf{Q}$ is clearly symmetric and positive semidefinite. Further, letting $R_i, \ i = 1, \ldots, n$, denote the stochastic return on asset i, the ijth element of $\mathbf{Q}$ is

$$Q_{ij} = \frac{\text{Cov}(R_i, R_j)}{(\mu_i - r_b)(\mu_j - r_b)} = \text{Cov}\left(\frac{R_i}{\mu_i - r_b}, \frac{R_j}{\mu_j - r_b}\right).$$

The purpose of this transformation is to capture a large portion of the variation of the objective of (MV) in only a few eigenvalues and eigenvectors. Figure 13 shows that the eigenvalue structure of the matrix Σ is altered considerably as we change the value of r_b. Some preliminary experiments led to the conclusion that only values of r_b that are of the same magnitudes as the expected portfolio returns are of interest.

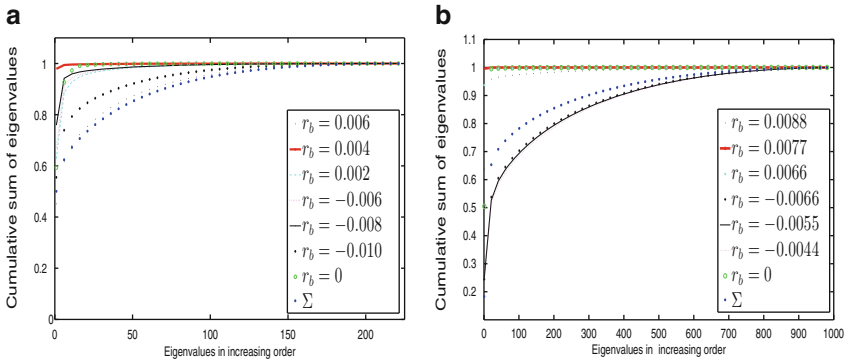


Fig. 13 Cumulative eigenvalue distribution for the matrix $\mathbf{Q}$ with different r_b values, compared to the matrix Σ. (**a**) For 225 assets. (**b**) For 1,000 assets

Figure 13a shows that more than 99 % of the cumulative sum of the eigenvalues for matrix $\mathbf{Q}$ is captured in less than 5 eigenvalues for $r_b = 0.004$, while about 150 eigenvalues are required to capture the same cumulative sum for the matrix Σ. From Fig. 13b, most cumulative sum of eigenvalues is captured within a few eigenvalues when using $r_b = 0.0077$. Note that some values of r_b can give the matrix $\mathbf{Q}$ a more dispersed eigenvalue structure compared to Σ. The behaviour is similar for the 500 asset instance.

The intuition we get from Fig. 13 is that the objective function in (4) can capture a larger portion of the variation in the objective function of (MV) with only a few eigenvalues and eigenvectors if matrix Σ is replaced with $\mathbf{Q}$. Problems (3) and (MV) can then be well approximated along the directions of a few largest eigenvalues.

Transformation of the model (MV) using the change of variable (12) leads to the following new model:

$$\begin{aligned} \min_{\mathbf{y}\in\Re^n} \quad & \mathbf{y}^{\mathrm{T}}\mathbf{Q}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{e}^{\mathrm{T}}\mathbf{y} \geq \mu_P - r_b \\ & \mathbf{D}\mathbf{y} \in \mathcal{S} \end{aligned} \tag{13}$$

Since matrix $\mathbf{Q}$ is also a covariance matrix, it can, like Σ, be decomposed into eigenvalues and eigenvectors. Following similar arguments as those used in Sect. 2.1, we get an approximate problem for (13), whose optimal value we denote by T_I^* for a subset I of eigenvalues and eigenvectors. If $\mathbf{y}_I^*$ is an optimal solution of such an approximate problem, then similarly an upper bound for V^* becomes

$$\hat{T}_I^* = \mathbf{y}_I^{*\mathrm{T}} \boldsymbol{Q} \mathbf{y}_I^*.$$

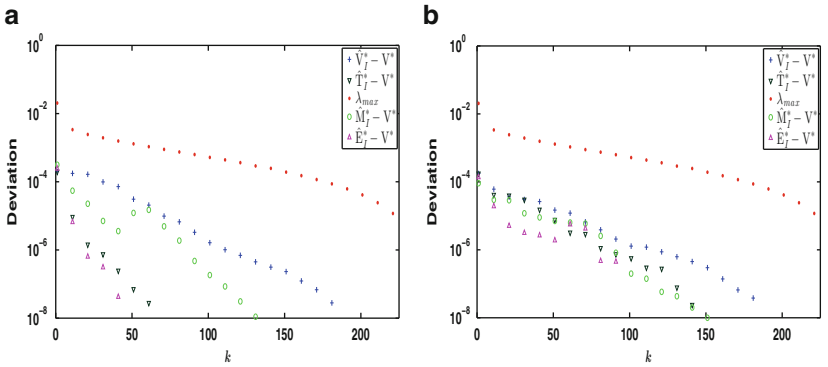


Fig. 14 Comparison of upper bounding quality between the transformed and untransformed model for 225 assets. (**a**) Instance 225A. (**b**) Instance 225B

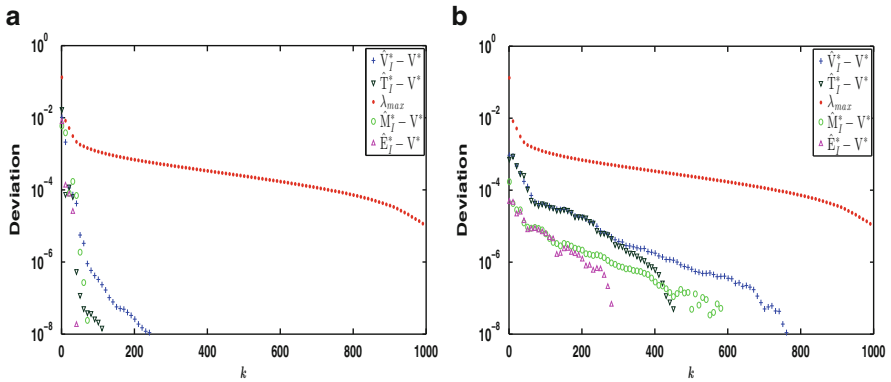


Fig. 15 Comparison of upper bounding quality between the transformed and untransformed model for 1,000 assets. (**a**) Instance 1000A. (**b**) Instance 1000B

Again, still following similar arguments as those used in Sect. 3.1 to get model (6), a model with a linearized term for the transformed model (13) can be constructed. We denote its optimal value by E_I^* for a chosen subset I of eigenvalues and eigenvectors. The upper bound of such a transformed model shall be denoted as $\hat{E}_I^*$. Below we present computational illustrations of the transformed model. We use the same problem instances as in Sect. 4.2, and the results obtained therein are compared to those obtained using the transformed model. The results are shown in Figs. 14 and 15, which are compared to Figs. 4 and 6, respectively.

For problem instance 225A, the transformed models with and without a linearized error term both perform better than even the untransformed model with a linearized error term. Although the transformed model without an error term in Fig. 14b for instance 225B performs relatively the same as the untransformed with a linearized error term, the transformed model with a linearized error term performs better than the other cases.

The transformation also reduces the deviations from optimality for the 1,000 asset instances, as shown in Fig. 15. For the low-risk scenario, exact solutions can be obtained with less than 30 % of the total number of eigenvalues and eigenvectors, as compared to almost 70 % for the untransformed model.

The intuition we draw from using the transformation is that it can alter the eigenvalue distribution of the covariance matrix to a desired one such that the problem (MV) can be well approximated using fewer largest eigenvalues.

6 Conclusion and Further Research

We have given a new insight into the mean-variance portfolio optimization problem which is based on performing a decomposition of the covariance matrix by means of its eigenvalues and eigenvectors. This decomposition amounts to restating the mean-variance problem in terms of uncorrelated eigenportfolios. When only a subset of the eigenportfolios is considered, we obtain a mean-variance problem that is a well-defined approximation of the original one. Our theoretical analysis and numerical illustrations reveal close relationships between the risk-return characteristic of the mean-variance problem under consideration, the cardinality of an optimal portfolio, and which, and how many, of the eigenportfolios that needed to be considered in order to well approximate the mean-variance problem.

According to the result of Corollary 5, the eigendecomposition enables the computation of near-optimal portfolios with controlled cardinalities. Further, as noted in Remark 2, an optimal solution to the approximate mean-variance problem (6) with cardinality K gives lower and upper bounds to a cardinality constrained mean-variance problem. These observations indicate that the eigendecomposition approximation can be employed for developing algorithms for the cardinality constrained mean-variance problem. We are currently exploring this possibility.

The numerical results presented in Sect. 5 are interesting and promising. A solid theoretical motivation for the transformation used, and an explanation for its effect on the eigenvalue distribution and the performance of model (6), is however lacking. This is an intriguing subject for further research.

We are presently investigating the use of the approximate model (6) in an iterative manner, as a vehicle for solving the mean-variance problem exactly. Another opportunity for further research is the extension of the eigendecomposition approach to other classes of quadratic programming problems, including cardinality constrained quadratic programs.

Acknowledgements The authors would like to acknowledge the Eastern African Universities Mathematics Programme (EAUMP) and the International Science Programme (ISP) at Uppsala University, for the financial support.

References

1. Avellaneda, M., Lee, J.H.: Statistical arbitrage in the us equities market. Quant. Finan. **10**(7), 761–782 (2010)
2. Bai, J., Ng, S.: Determining the number of factors in approximate factor models. Econometrica **70**(1), 191–221 (2002)
3. Bai, J., Ng, S.: Principal components estimation and identification of static factors. J. Econ. **176**(1), 18–29 (2013)
4. Bai, J., Shi, S.: Estimating high dimensional covariance matrices and its applications. Ann. Econ. Finance **12**(2), 199–215 (2011)
5. Broadie, M.: Computing efficient frontiers using estimated parameters. Ann. Oper. Res. **45**(1–4), 21–58 (1993)
6. Carhart, M.M.: On persistence in mutual fund performance. J. Finance **52**(1), 57–82 (1997)
7. Chopra, V.K., Hensel, C.R., Turner, A.L.: Massaging mean-variance inputs: returns from alternative global investment strategies in the 1980s. Manag. Sci. **39**(7), 845–855 (1993)
8. Cragg, J.G., Donald, S.G.: Inferring the rank of a matrix. J. Econ. **76**(12), 223–250 (1997)
9. Fama, E.F.: Multifactor portfolio efficiency and multifactor asset pricing. J. Financ. Quant. Anal. **31**(4), 441–465 (1996)
10. Fama, E.F., French, K.R.: Common risk factors in the returns on stocks and bonds. J. Financ. Econ. **33**(1), 3–56 (1993)
11. Fan, J., Fan, Y., Lv, J.: High dimensional covariance matrix estimation using a factor model. J. Econ. **147**(1), 186–197 (2008)
12. Forni, M., Reichlin, L.: Let's get real: a factor analytical approach to disaggregated business cycle dynamics. Rev. Econ. Stud. **65**(3), 453–473 (1998)
13. Gao, P., Huang, K.: Aggregate consumption-wealth ratio and the cross-section of stock returns: some international evidence. Ann. Econ. Finance **9**(1), 1–37 (2008)
14. Kolm, P.N., Tütüncü, R., Fabozzi, F.J.: 60 years of portfolio optimization: practical challenges and current trends. Eur. J. Oper. Res. **234**(2), 356–371 (2014)
15. Larsson, T., Migdalas, A.: An algorithm for nonlinear programs over Cartesian product sets. Optimization **21**(4), 535–542 (1990)
16. Ledoit, O., Wolf, M.: Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. J. Empir. Finance **10**(5), 603–621 (2003)
17. Lettau, M., Ludvigson, S.: Resurrecting the (c)capm: a cross-sectional test when risk premia are time varying. J. Polit. Econ. **109**(6), 1238–1287 (2001)
18. Lintner, J.: Security prices, risk, and maximal gains from diversification. J. Finance **20**(4), 587–615 (1965)
19. Markowitz, H.: Portfolio selection. J. Finance **7**(1), 77–91 (1952)
20. Merton, R.C.: An intertemporal capital asset pricing model. Econometrica **41**(5), 867–887 (1973)
21. Mossin, J.: Equilibrium in a capital asset market. Econometrica **34**(4), 768–783 (1966)
22. OR-Library: https://files.nyu.edu/jeb21/public/jeb/orlib/portinfo.html (2014)
23. Sharpe, W.F.: A simplified model for portfolio analysis. Manag. Sci. **9**, 277–293 (1963)
24. Steuer, R.E., Qi, Y., Hirschberger, M.: Comparative issues in large-scale meanvariance efficient frontier computation. Decis. Support Syst. **51**, 250–255 (2011)
25. Wu, J., Yang, H.: Two stochastic restricted principal components regression estimator in linear regression. Commun. Stat. Theory Methods **42**(20), 3793–3804 (2013)

Three Aspects of the Research Impact by a Scientist: Measurement Methods and an Empirical Evaluation

Boris Mirkin and Michael Orlov

Abstract Three different approaches for evaluation of the research impact by a scientist are considered. Two of them are conventional ones, scoring the impact over (a) citation metrics and (b) merit metrics. The third one relates to the level of results. It involves a taxonomy of the research field, that is, a hierarchy representing its composition. The impact is evaluated according to the taxonomy ranks of the subjects that have emerged or have been crucially transformed due to the results by the scientist under consideration Mirkin (Control Large Syst Spec Issue 44:292–307, 2013). To aggregate criteria in approaches (a) and (b) we use an in-house automated criteria weighting method oriented towards as tight a representation of the strata as possible Orlov (Bus Inf, 2014). To compare the approaches empirically, we use publicly available data of about 30 scientists in the areas of data analysis and machine learning. As our taxonomy of the field, we invoke a corresponding part of the ACM Computing Classification System 2012 and slightly modify it to better reflect results by the scientists in our sample. The obtained ABC stratifications are rather far each other. This supports the view that all the three approaches (citations, merits, taxonomic rank) should be considered as different aspects, and, therefore, a good method for scoring research impact should involve all the three.

Keywords Evaluation of research impact • Citation index • Merit metrics • Aggregate criteria • Linstrat method • Multicriteria analysis

B. Mirkin (✉)
School of Computer Science and Information Systems, Birkbeck, University of London, London, UK

Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation
e-mail: mirkin@dcs.bbk.ac.uk

M. Orlov
Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_12

1 Introduction: The Problem and Background

The issue of measuring research impact is attracting intense attention of scientists because metrics of research impact are being widely used by various managing bodies and by public at large as easy-to-get shortcuts for judging of comparative strengths among scientists, research centers, and universities. The citation index and such its derivatives as Hirsch index are produced by a number of organizations including the inventors, currently named Thomson Reuters [32], and Google. These indexes are used sometimes in evaluation and management in sciences, which can be subject to debate because of over-simplifications immanent to bibliometrics [2]. There have been a number of proposals to amend the indexes, say, by using less extensive characteristics, such as centrality indexes in the intercitation graphs [5] or by following only citations in "lead scientists" work [4], see also [8]. Other proposals deny the usefulness of bibliometrics altogether; some propose such drastic measures as the "careful socialization and selection of scholars, supplemented by periodic self-evaluations and awards" [25], that is, moving back to the closed orders of monk-scientists. Other, more practical systems, such as the UK Research Assessment Exercise (RAE, recently rebranded as REF) intends to assess most significant contributions only, and in a much informal way, which seems a better option. Yet there have been criticisms of the RAE-like systems as well: on the one hand, in the absence of a citation index, the peer reviews do not manifest any consistency in evaluations [1, 10], and, on the other hand, in the long run, the system has cut off everything which is out of the mainstream [17]. Therefore, a recent initiative by a group of influential scientists DORA [28], while rejecting the bibliometrics as the only assessment source, proposes to switch from counting publications only to checking for the whole list of scientific production including data sets, patents, and codes among others. The US National Science Foundation already modified its instructions so that the outputs of scientific research include products rather than just publications [28]. This goes in line with what Alfred Nobel, the founder of the most prestigious science prize, has expressed in his will: the prize goes to those who "have conferred the greatest benefit on mankind" which is further detailed, say for physics, as " have made the most important discovery or invention within the field of physics" [22].

We adhere to this opinion. This chapter is an attempt at exploring aspects of the concept of larger than papers researcher's productivity. Looking from a practical side, one can recognize that currently there are at least four types of products of scientific research:

1. producing novel scientific results to be described in papers and monographs;
2. participating in the organization of sciences such as being a journal editor or running a research conference;
3. transferring knowledge to and training of younger generations such as undergraduate and postgraduate students;
4. developing technology innovations including patents and other industry-related products.

They all should be counted as parts of the impact by a scientist.

Therefore, we are going to explore how these can be reasonably measured and aggregated to derive a reasonable measure of research impact. We recognize the difficulties in measuring the last item, of technical innovations, for the currently living scientists because not so many of them ever get patents. To justfully abandon this item we restrict ourselves with university-based researchers only, since academics normally are not required to get a practical use of their research results.

Another issue is in finding a direct measure to score the research results, item 1, which is so remarkably avoided by using bibliometrics instead. Here we are going to employ a recently proposed idea of using a hierarchical taxonomy of a research field for mapping research results in the field to those subjects that have been created or drastically revised in the light of these results. The ranks of the receiving nodes define the rank of the research results [19].

Another innovation reported in this chapter is in the way of combining multiple criteria. A number of popular approaches to multicriteria rank aggregation rely on weighted combinations of criteria in such a way that the weights are defined either manually or in a supervised manner. For example, the former applies to computing university league tables, and the latter is characteristic for defining ABC classifications of inventory items. Automatically deriving the weights has been pursued as well, mostly in the format of the eigenvector corresponding to the maximum eigenvalue for a similarity-between-criteria matrix such as RankClus [26, 29] and PCA [18]. This approach is much relevant when the criteria are well correlated so that a better entity over one criterion would be better over most other criteria. If, however, criteria are essentially conflicting at different entities, the first eigenvector would take into account too little of the data scatter and, therefore, may be somewhat inappropriate. We develop an approach which is adequate at both correlated and conflicting criteria. According to our approach, the issue is to be solved by finding such a direction in the criteria space that all the entities are projected into compact well-separated clusters on it so that the orthogonal hyperplanes may be considered as boundaries between different multicriterial strata of entities. This approach was introduced and substantiated recently in [23, 24].

One more innovation described here is a case of practical implementation of our approaches. To be specific, we focus on the field of Computer Science related to data analysis, machine learning, cluster analysis, and data mining. As a relevant taxonomy of the domain we take relevant parts of the ACM Computing Classification System 2012 [30]. We pick up a sample of 30 leading scientists in the field such that the information of their research results is publicly available. We consider three sets of criteria for research contributions: (a) one comprises three Google citation criteria, (b) the second, criteria for items of merit, 2 and 3 from the list above, and (c) the third utilizes adjusted ranks of research results within the taxonomy.

Our preliminary hypothesis is that the aggregate scales of both (a) citation and (b) merit relate to popularity of scientists rather than anything else. Therefore, the combined scales for (a) and (b) should have a rather high correlation between them. On the other hand, the level of results has no straightforward relation to

popularity—the latter much depends on the scientist's character and communication skills, while the former, on talent and luck. So any reasonable scale of the level of results should have rather low correlation with both citation index and merit index. Our computations do show that this is largely true at our data, although the level of correlation between (a) and (b) is not that high. To an extent, this observation supports the views expressed in DORA declaration [28]. Also, we may conclude that our method of mapping research results to a taxonomy of the field (MMRRTF) could be considered a good way forward. It does involve a great deal of manual component, of course. However, it is based on an agreed upon taxonomy of the domain and explicitly mapping the results to taxonomy nodes. Therefore, its results are explicitly expressed and admit public discussions of them, which leads to much less inconsistency in the assessments than just mere subjective evaluations by panel members.

The remainder is organized as follows. The next section provides an algorithmic background for the Linstrat method for aggregating criteria in the format of a weighted sum of them [23, 24]. Our method for mapping research results to a taxonomy of the fields is presented there too. Section 3 describes how our sample of scientists has been formed and how scientists' ranks have been defined by adapting an extract from the taxonomy in ACM CCS [30]. Section 4 presents data related to features of (a) citation and (b) merit for our sample. Our results in determining stratifications and criteria weights are presented here as well. Section 5 concludes with a summary and future work directions.

2 Methodology

2.1 The Problem of Stratification

There is a general understanding that in the ranking problem one usually looks for an ordered partition in which entities in the same class are considered to be equivalent over a pre-specified set of criteria, rather than for just a linear ordering of the entities. Reasons for this may include a degree of indifference of the decision makers (as reflected, say, in the concept of ABC ranking in inventories) or a degree of imprecision in the measurement of criteria or both. We refer to a partition, classes of which are linearly ordered by a relation of precedence, as a stratification. Such areas as sociology and mineralogy use this term exactly in this sense to express social inequality in the former and depth/time precedence in the latter.

Consider an example. Table 1 contains normalized food and housing prices for a foreigner in 10 cities of the world [7].

The left part of Fig. 1 presents a three cluster partition found using k-means clustering method with cities Copenhagen, New York, and Peking taken as the initial centers. The right part of Fig. 1 presents a three strata stratification corresponding to the direction of a combined criterion $F = 0.4789 * HousingP + 0.5211 * FoodP$. This combined criterion can be interpreted as a measure of "cost of living" that takes into account the difference in the relative importance of the criteria.

Table 1 Prices of housing and food for a foreigner in ten cities normalized so that the minimum is zero and maximum, the hundred

City	Housing	Food
Moscow	96.7284	56.0364
London	93.2099	62.4146
Tokyo	100.0000	44.4191
Copenhagen	42.7160	100.0000
New York	96.7284	38.9522
Peking	59.9383	12.0729
Sydney	34.4444	19.5900
Vancouver	12.9630	10.2506
Johannesburg	0	5.2392
Buenos Aires	14.1975	0

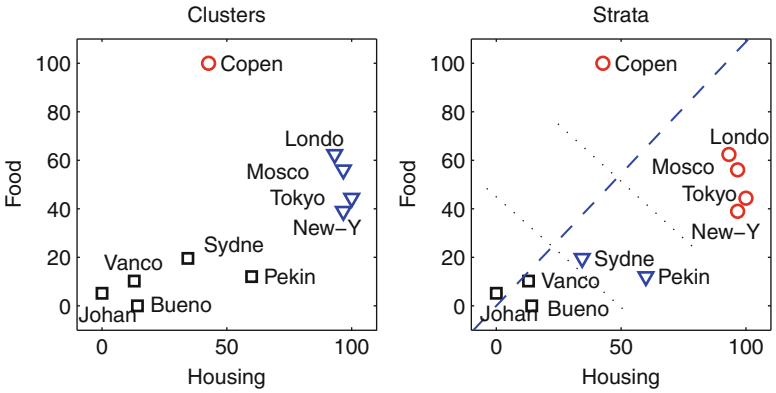


Fig. 1 Ten cities over two normalized criteria: Housing price and Food price. They are partitioned in three clusters (on the *left*) and in three strata (on the *right*)

As expected, clusters consist of similar cities (see Fig. 1 on the left). Those labeled by a square have relatively low prices for both foods and housing. Cluster labeled by a circle is a singleton consisting of just Copenhagen, with a highest food price and moderate housing prices. The cluster of triangles on the right, in contrast, is of highest housing prices and moderate food prices. The strata, on the right side, are organized over a different principle. The first stratum, for example, is not a cluster but rather a Pareto boundary at highest prices. Each of the remaining cities is dominated, over both criteria, by a city from the first stratum. It is formed not according to similarity but rather according to the combined weighted criterion as a set of a higher cost of living. The second stratum is a set of a moderate living cost, and the third, of the lowest living cost in the set.

One can classify methods for multicriteria stratification according to the extent of the assumed elasticity of the criteria to each other or the value trade-off [15]. A constant elasticity e of criterion f_1 towards criterion f_2 would mean that a change of criterion f_2 by a unity is equivalent to the opposite change of f_1 in e units, independently of values of these and other criteria. That is, criteria f_1 and f_2 can be

combined into weighted sum $f_1 + ef_2$ in this case. The case of a constant elasticity between all the criteria $f_1(x), f_2(x), \ldots, f_m(x)$ assumes that they can be equivalently substituted by an aggregate criterion $f(x)$ which is expressed as their weighted sum $f(x) = w_1 f_1(x) + w_2 f_2(x) + \ldots + w_m f_m(x)$, where $w_1, w_2, \ldots, w_m$ are non-negative constant weight coefficients summing to 1.

An opposite case is when all the criteria are mutually incomparable and there is no way that a change in one criterion can be equivalently represented by a change in another criterion. That is, each criterion must be taken into consideration whatever the other criteria values are. The absence of interrelation among criteria leads to the multivariate relation "better than," that is, "better over every single criterion," and the concept of Pareto boundary as the only solution that needs no interrelation between criteria at all. Yet there is a kind of equivalence between these two extremes: under rather mild mathematical conditions on the criteria and the sets at which they are defined, every x maximizing the combined criterion $f(x) = \sum_{t=1}^{m} w_t f_t(x)$ does belong to the Pareto boundary. And vice versa, any point x belonging to the Pareto boundary can be found as a maximizer of the combined criterion $f(x) = \sum_{t=1}^{m} w_t f_t(x)$ for some x-specific set of weights w (see Fig. 2).

For a detailed review of various interpretations of criteria weight coefficients one may refer to [9]. Much work on multicriterion ranking has been done along the lines of using an external information, say from a Decision Maker, to try to reveal as much information on comparability of criteria at various preference profiles (see, for example, Electre method [12] or PROMETHEE method [6]). Papers [21, 27] develop methods for dividing resources in ABC groups according to their importance for the company by using a criteria weighting system. The groupings are determined by using a combined weighted criterion in which weights are found by solving a linear programming problem. These weights are not constant but depend on the variants being compared.

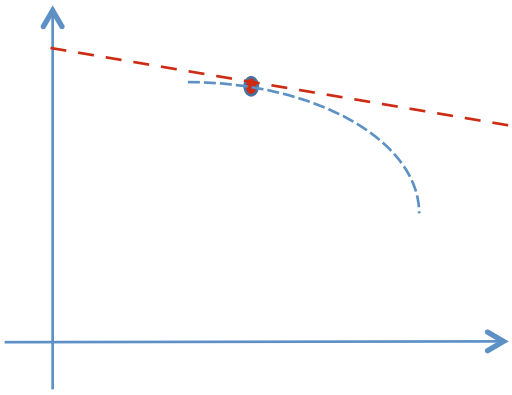


Fig. 2 An illustration of the equivalence between two approaches; one of weighted combined criteria and the other, of Pareto boundary solutions

As we concentrate on the case of a weighted combined criterion with constant weights, we should mention the following. In the real world, there are some applications in which weighted combined criteria are used in such a way that the weights are chosen manually by experts; such are methods applied in composition of university league ranking tables (see, for instance, [31]). In some works, weights are learned in a supervised or semi-supervised manner [16].

2.2 Linstrat Criterion and Method

We think of our Linstrat method as that inspired by the idea that Pareto boundaries, formed by consecutive "shaving" off the current Pareto boundary from the data set, can be approximated as strata between parallel hyperplanes whose normal vector, that is, the vector of criteria weights, is taken such that the projections of entities under consideration within each stratum are as close to each other as possible. This idea leads to an optimization problem described below.

Consider a set of N items evaluated over M criteria so that the evaluation scores can be represented as a matrix (x_{ij}), where $i \in 1,\ldots,N$ are the items or actions, $j \in 1,\ldots,M$ criteria, and x_{ij} is the value of jth criterion at the ith item. Assume some criteria weights $w = (w_1, w_2, \ldots, w_M)$ such that $w_j \geq 0$ at every j and $\sum_j w_j = 1$. These weights are taken into account in the combined criterion $f = \sum_{j=1}^{M} w_j x_j$ where x_j is jth column of matrix $X = (x_{ij})$. The problem is to divide the item set in K disjoint subsets $S = \{S_1, ..S_k, \ldots, S_K\}, k = 1,\ldots,K$ referred to as strata, according to values of the combined criterion f. Each stratum is characterized by a value of the combined criterion c_k, referred to as the stratum value, or center. These values are ordered so that $c_k > c_l$ whenever $k < l$. That means that any item from kth stratum is ranked higher, or is more preferable, than any item from stratum l if $k < l$.

Geometrically, strata are formed by layers between parallel planes in the space of criteria. At any stratum S_k, we assume that the value of the combined criterion $f_i = \sum_{j=1}^{M} w_j x_{ij}$ at any $i \in S_k$ approximates the stratum value c_k as much as possible. That is, in the equation $x_{i1}w_1 + x_{i2}w_2 + \ldots + x_{iM}w_M = c_k + e_i$, e_i is an error to be minimized over unknown weights w. The problem of finding an optimal w can be formulated as the following optimization problem with respect to weights w, centers $\{c\}$, and partitions S:

$$\begin{aligned} \min_{w,c,S} \quad & \sum_{k=1}^{K} \sum_{i \in S_k} \left(\sum_{j=1}^{M} x_{ij} w_j - c_k\right)^2 \\ \text{such that} \quad & \sum_{j=1}^{M} w_j = 1 \\ & w_j \geq 0, j \in 1 \ldots M. \end{aligned} \tag{1}$$

At any given weight vector w, the criterion in (1) is the conventional square-error clustering criterion of K-means clustering algorithm over a single feature, the combined criterion $f = \sum_{j=1}^{M} w_j x_j$. This implies that finding the optimal stratification S, at a pre-specified K, amounts to finding $K-1$ points dividing the f-axis in K intervals to minimize the within-cluster variance, and the optimal centers c_k are just within-cluster means of f. An optimal stratification over a single feature can be found by using Fisher's dynamic programming clustering algorithm [13]. Therefore, the difficulty in the minimization of (1) is concentrated in the task of finding an appropriate w at a given stratification S. If an algorithm for this is specified, then one can proceed in the manner of an alternating minimization algorithm: starting from some weight vector $w(0)$, find optimal S and c. Based on these, find an appropriate weight vector $w(1)$, etc.

At first, we used an evolutionary algorithm for minimizing (1) with respect to w at a given S and c. However, such an algorithm as a whole leads to unstable solutions at some data sets and, moreover, the solutions at times are inferior to those found by using other approaches [20]. A modification based on a direct algorithm for solving the quadratic programming problem is proposed in [23]. It starts from a random w, but leads to a stable solution in most cases. Moreover, in our experiments with synthetic data sets it typically outperforms its competitors by a high margin [23, 24]. Therefore, we use this version of Linstrat through the entire material reported in this chapter.

2.3 Taxonomic Rank of a Scientist

The concept of taxonomic rank is not uncommon in the sciences. Moreover, it is quite popular in biology: "A Taxonomic Rank is the level that an organism is placed within the hierarchical level arrangement of life forms.", according to a dictionary (see http://carm.org/dictionary-taxonomic-rank). Say, *Eukaryota* is a domain (rank 1) containing *Animals* kingdom (rank 2). The latter contains *Cordata* phylum (rank 3) which contains *Mammals* class (rank 4) which contains *Primates* order (rank 5) which contains *Hominidae* family (rank 6) which contains *Homo* genus (rank 7) which contains, at last, *Homo sapiens* species (rank 8).

According to the proposal in [19], the taxonomic rank of a scientist should be defined in a similar way. The relevant science domain should be structured by a hierarchical taxonomy such as that in Fig. 3. The rank of a scientist is defined then as the rank of a subdomain which has appeared because of the scientist's work or has been substantially transformed because of that. For example, if a domain has been structured as shown in Fig. 3 and a scientist's work has highly affected the subdomain labeled as A.1.2 (see the triangle indicating that), then her/his rank would be 3, the number of characters, other than dot, in the code of the subdomain. Of course, this goes in the opposite direction: the higher the rank, the lower the level.

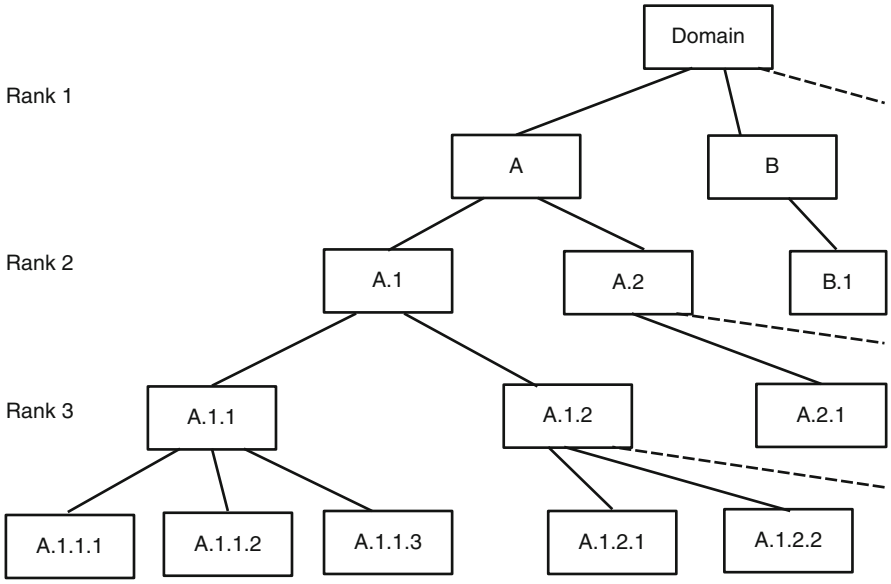


Fig. 3 An illustrative taxonomy of a domain. The *triangle* shows that subdomain A.1.2 has been seriously affected by the results in example

In a practical implementation, when scoring the level of results for currently living scientists, it is much easier to map their individual papers to the taxonomy rather than the overall achievements. Indeed, the overall achievement is not easy to formulate, whereas an individual paper usually represents a single individual achievement which is not difficult to map to the taxonomy, even if onto two or more subdomains. Together with the plurality of one's results, this leads to the issue of multiple subdomains developed or transformed by a scientist. If the work of a scientist has affected a number of subdomains in a taxonomy, what rank should be assigned to the contribution her/him?

It seems natural that the contribution of an achievement at a lower layer to that of the highest layer achievement is less by an order of magnitude at scoring the taxonomic rank of a scientist. Therefore, of all the levels of the taxonomic hierarchy affected by them first and foremost the highest level is to be used. In the case that only one subdomain is considered as highly affected by the scientist, then her/his rank is defined as the taxonomy layer to which the subdomains belong. Such is the case illustrated in Fig. 3 if the subdomain in question is A.1.2, then the scientist's rank is 3. In the case when two or more subdomains on the highest level are affected by a scientist, the rank should be further decreased within the unit interval separating the current rank from the higher one. The scale of the drop should depend on the range of number of possibly affected subdomains. In our empirical investigation, we considered, for each of the scientists in our sample, at most five papers leading to ground-breaking discoveries or methods within the taxonomy. Thus, we thought that

each additional subdomain of the highest level affected should make a drop in the rank equal to 0.1. Then, an additional drop caused by a node of a lower layer should be about 0.01. For example, if a scientist's results highly affected 4 subdomains of rank 4 and 3 subdomains of rank 3, then the taxonomy rank of the scientist will be 2.76. Indeed, 4 subdomains of rank 4 contribute -0.01 each; one affected subdomain of rank 3 leads to the rank value 3, and each of the two remaining rank 3 subdomains decreases that by 0.1 so that the final rank is $3-2*0.1-4*0.01=2.76$. To make it simpler, we can assume that additionally 0.1 is subtracted from each of the ranks found—this will not affect the results of the data normalization to 0–100 scale, but the formula for computing the rank gets very simple. To formulate it, let us denote R the set of nodes assigned to a scientist. Let it be partitioned in subsets R_h, $h \in H$, of the same rank where $H=(h_1,h_2,\ldots,h_p)$ and $h_1<h_2<\ldots<h_p$. Then the taxonomic rank of the scientist is defined as

$$r = h_1 - \sum_{k=1}^{p} (0.1)^k * h_k.$$

This method for assigning a scientist her/his taxonomic rank suffers of issues of which the following three seem of importance. First, the method is not automated. The mapping of a research paper to the taxonomy is done manually, so that the result is highly affected by the person(s) performing the mapping; it depends on both the knowledge of the domain and its history as well as on the extent of understanding of the result. Still, any mapping decision is an explicitly stated judgement which can be discussed openly and corrected if needed. What is important is that the subjective part in the decision is quite minor. This much differs from the currently used method of peer-reviewing. Indeed, peer-based results can be highly subjective and dependent on various external features such as citation scores [10, 11, 33]. Second, there can be no regular service for updating the taxonomy of the domain. In this case, a ground-breaking paper can be assigned to a wrong subdomain just because the proper one is not yet present in the taxonomy under consideration. In our assignments reported in the next section this did happen more than once. In such cases, because of the presence of the senior coauthor whose career spans for the past 50 years, we did not hesitate to expand the taxonomy with updated subdomains. This means that this drawback can be dealt with, at least partly. Third, and foremost, unlike in biology, the taxonomies of specific research domains, especially those being under development, are subject to debate. Some popular concepts may go in a few years, some new concepts may emerge, some new links can be discovered, whereas some old links become obsolete. This is especially true for such a dynamic area as computer-related computations and services in which the theoretical thinking is highly affected by the industrial progress in hardware. Say, initially computers were oriented at computations, then at data processing, and nowadays, it seems they are oriented at networking. Change in the overall perspective necessarily leads to a drastic change in the taxonomy of the domain. For example, if one compares the current ACM Computing Classification System 2012 [30] with its previous version, the ACM Classification of Computing Subjects 1998, one cannot help

but notice great differences in both the subdomain list and the structure of their mutual arrangement. Yes indeed, the current taxonomies of domains can be not well structured and, thus, unstable. However, the appreciation of the level of results goes in line with the taxonomic structure of the domain. The more important is a subdomain, the more important are ground-breaking results of it. Indeed, unlike the level of citations, the recognition of the relative importance of this or that subdomain is subject to change. This just shows that the domain taxonomy cannot be considered stable while the domain is being developed, so is the level of results.

3 Developing an Empirical Testing Base for the Taxonomic Rank Evaluation

To put a testing to our methods we need, first of all, to take a sample of scientists working in the same domain and score their contributions. The following steps should suffice:

1. Specify a knowledge domain
2. Take its appropriate taxonomy
3. Collect a representative sample of scientists with results in the domain
4. For each of the scientists in the sample, map her/his ground-breaking results to the taxonomy
5. Compute the taxonomic rank of each of the scientists in the sample

Further on we describe our work on implementation of these steps.

3.1 A Taxonomy of the Data Analysis Subjects

For an empirical evaluation, we decided to focus on the domain of intelligent data analysis including what is referred to as machine learning and data mining areas. We know some of its history and the current state. We feel that our expertise in other domains is even more embryonic. As to the taxonomy of the domain, we tried first to consider taxonomy from textbook [18], then from textbook [14]—both appear to be difficult to use for mapping individual research results into because both cover rather basic subjects only, and it remains entirely unclear at which places in them real-world research results should be mapped to. In this aspect, the ACM CCS 2012 taxonomy has provided us with much better guidance. Parts of ACM CCS 2012 related to the domain under consideration can be considered as composed of the branches in the ACM CCS presented in Tables 2 and 3.

This part extended by the less general concepts from ACM CCS 2012 is presented in Table 3. For the sake of saving room, parts of the hierarchy not affected by the mapping of research results are minimized. On the other hand, the part under consideration is updated by adding items concerning the outstanding

Table 2 ACM CCS 2012 high rank items covering data analysis, machine learning, and data mining

Subject index	Subject name
1.	Theory of computation
1.1.	Theory and algorithms for application domains
2.	Mathematics of computing
2.1.	Probability and statistics
3.	Information systems
3.1.	Data management systems
3.2.	Information systems applications
3.3.	World Wide Web
3.4.	Information retrieval
4.	Human-centered computing
4.1.	Visualization
5.	Computing methodologies
5.1.	Artificial intelligence
5.2.	Machine learning

results by scientists from our sample that have not been covered in the taxonomy. These concern, as a rule, only leaves of the tree, as can be seen in Table 3. This table represents that part of the taxonomy which has been used for mapping there outstanding results by scientists from our sample. The subdomains (taxonomy nodes) affected by these results are marked by one or two stars. A one star node refers to a subdomain being part of ACM CCS 2012; a two star node refers to a subdomain added by the authors.

3.2 Sample of Scientists and Their Taxonomic Ranks

In our sampling, we rely on Google citation indexes and try to pick up those with maximum citations. Ideally, we wanted to take about 15–20 scientists from the USA and a couple of scientists from a country such as Australia, Canada, China, France, Germany, Netherlands, Russia, and the UK so that the relative contributions by countries would be reflected in the sample. This also would warrant a variation in citation levels: from many dozen thousands at some of the USA scientists to a very few thousands at those in Europe. This ideal composition, though, was difficult to achieve because for any scientist from the sample we needed data not only on citation and taxonomic rank but on merit as well. The merit data were not always available so that we went as far as to contact over e-mail those of sampled scientists for whom the merit data were not easily available, asking them to fill in the slots of

Table 3 ACM CCS 2012-based taxonomy of data analysis, machine learning, and data mining

Subject index	Subject name
1.	Theory of computation
1.1.	Theory and algorithms for application domains
1.1.1.	Machine learning theory
1.1.1.1.	Sample complexity and generalization bounds
1.1.1.2.	Boolean function learning
1.1.1.3.*	Unsupervised learning and clustering
1.1.1.4.	Kernel methods
1.1.1.4.1.	Support vector machines
1.1.1.4.2.	Gaussian processes
1.1.1.4.3.**	Modelling
1.1.1.5.	Boosting
1.1.1.6.*	Bayesian analysis
1.1.1.7.– 1.1.2.12.	···
2.	Mathematics of computing
2.1.	Probability and statistics
2.1.1.	Probabilistic representations
2.1.1.1.	Bayesian networks
2.1.1.2.*	Markov networks
2.1.1.3.– 2.1.1.8.	···
2.1.1.8.1.	Kernel density estimators
2.1.1.8.2.	Spline models
2.1.1.8.3.*	Bayesian nonparametric models
2.1.2.	Probabilistic inference problems
2.1.2.1.– 2.1.3.6.	···
2.1.3.7.	Kalman filters and hidden Markov models
2.1.3.7.1**	Factorial HMM
2.1.3.8.– 2.1.5.3.	···
2.1.5.3.1.*	Robust regression
2.1.5.4.– 2.1.5.10.	···
2.1.6.– 2.1.9.	···
3.	Information systems
3.1.	Data management systems
3.1.1.	Database design and models
3.1.1.1.– 3.1.1.5.	···
3.1.1.5.2.*	Data streams
3.1.1.5.3.– 3.1.1.5.7.	···
3.1.2.	Data structures
3.1.2.1.	Data access methods
3.1.2.1.1.*	Multidimensional range search
3.1.2.1.2.– 3.1.2.1.5.	···
3.1.2.2.– 3.1.5.9.	···

(continued)

Table 3 (continued)

Subject index	Subject name
3.2.	Information systems applications
3.2.1.	Data mining
3.2.1.1.	Data cleaning
3.2.1.2.	Collaborative filtering
3.2.1.2.1**	Item-based
3.2.1.2.2**	Scalable
3.2.1.3.*	Association rules
3.2.1.3.1**	Types of association rules
3.2.1.3.2**	Interestingness
3.2.1.3.3**	Parallel computation
3.2.1.4.	Clustering
3.2.1.4.1**	Massive data clustering
3.2.1.4.2**	Consensus clustering
3.2.1.4.3**	Fuzzy clustering
3.2.1.4.4**	Additive clustering
3.2.1.4.5**	Feature weight clustering
3.2.1.4.6**	Conceptual clustering
3.2.1.4.7**	Biclustering
3.2.1.5.	Nearest-neighbor search
3.2.1.6.*	Data stream mining
3.2.1.7**	Graph mining
3.2.1.7.1**	Graph partitioning
3.2.1.7.2**	Frequent graph mining
3.2.1.7.3**	Graph based conceptual clustering
3.2.1.7.4**	Anomaly detection
3.2.1.7.5**	Critical nodes detection
3.2.1.8.**	Process mining
3.2.1.11**	Text mining
3.2.1.11.1**	Text categorization
3.2.1.11.2**	Key-phrase indexing
3.2.1.10.**	Data mining tools
3.2.1.9**	Sequence mining
3.2.1.9.1.**	Rule and pattern discovery
3.2.1.9.2.**	Trajectory clustering
3.2.1.9.3**	Market graph
3.2.1.12**	Formal concept analysis
3.3.	World Wide Web
3.3.1.	Web mining
3.3.1.1.– 3.3.1.5.	···
3.3.1.6**	Knowledge discovery
3.4.	Information retrieval
3.4.1.	Document representation
3.4.1.1.– 3.4.1.5.	···

3.4.1.6.*	Ontologies
3.4.1.7.	Dictionaries
3.4.1.8.	Thesauri
3.4.2.– 3.4.3.	···
3.4.4.	Retrieval models and ranking
3.4.4.1.*	Rank aggregation
3.4.4.2.– 3.4.4.4.	···
3.4.4.5.*	Learning to rank
3.4.4.6.– 3.4.7.3.	···
4.	Human-centered computing
4.1.	Visualization
4.1.2.	Visualization techniques
4.1.2.1.– 4.1.2.6.	···
4.1.2.7**	Elastic maps
4.1.3.	Visualization application domains
4.1.3.1.–4.1.3.4.	···
4.1.4.– 4.1.7.	···
5.	Computing methodologies
5.1.	Artificial intelligence
5.1.1.	Natural language processing
5.1.1.2.– 5.1.1.7.	···
5.1.1.7.1**	Wikipedia based semantics
5.1.1.8.	Phonology/morphology
5.1.1.9.	Language resources
5.1.2.	Knowledge representation and reasoning
5.1.2.1.– 5.1.2.3.	···
5.1.2.4.*	Probabilistic reasoning
5.1.2.5.– 5.1.2.12.	···
5.1.3.	Computer vision
5.1.3.1.	Computer vision problems
5.1.3.1.1.	Interest point and salient region detections
5.1.3.1.2.	Image segmentation
5.1.3.1.3.– 5.1.3.1.10.	···
5.1.3.2.	Computer vision representations
5.1.3.2.1.	Image representations
5.1.3.2.1.1*	2D PCA
5.1.3.2.2.	Shape representations
5.1.3.2.3.	Appearance and texture representations
5.1.3.2.4.	Hierarchical representations
5.2.	Machine learning
5.2.1.	Learning paradigms
5.2.1.1.	Supervised learning
5.2.1.1.1.*	Ranking

(continued)

Table 3 (continued)

Subject index	Subject name
5.2.1.1.2.	Learning to rank
5.2.1.1.3.*	Supervised learning by classification
5.2.1.1.4.– 5.2.1.1.6.	Supervised learning by regression
5.2.1.2.	Unsupervised learning
5.2.1.2.1.*	Cluster analysis
5.2.1.2.2.*	Anomaly detection
5.2.1.2.3.*	Mixture modeling
5.2.1.2.4.	Topic modeling
5.2.1.2.5.	Source separation
5.2.1.2.6.	Motif discovery
5.2.1.2.7.*	Dimensionality reduction and manifold learning
5.2.1.2.7.1**	Graph embedding
5.2.1.2.7.2**	Supervised dimensionality reduction
5.2.1.3.– 5.2.2.6.	···
5.2.2.7.*	Semi-supervised learning settings
5.2.2.7.1.**	Kernel approach
5.2.3.	Machine learning approaches
5.2.3.1.	Classification and regression trees
5.2.3.1.1**	Parallel implementation
5.2.3.1.2**	Splitting criteria
5.2.3.1.3**	Model trees
5.2.3.2.	Kernel methods
5.2.3.2.1.**	Kernel support vector machines
5.2.3.2.1.1**	Dynamic kernel SVM
5.2.3.2.2.	Gaussian processes
5.2.3.2.3**	Kernel matrix
5.2.3.2.4**	Kernel independent components
5.2.3.2.5**	Kernel-based clustering
5.2.3.3.	Neural networks
5.2.3.3.1**	Self-organized map
5.2.3.3.2**	Training approaches
5.2.3.3.2.1**	Evolutionary approach
5.2.3.3.3**	Representation
5.2.3.3.3.1**	Rule-based network architecture
5.2.3.3.3.2**	Fuzzy representation
5.2.3.3.4**	Evolving NN
5.2.3.3.5**	Ensembling
5.2.3.4.	Logical and relational learning
5.2.3.4.1.	Inductive logic learning
5.2.3.4.2.	Statistical relational learning
5.2.3.5.*	Learning in probabilistic graphical models

5.2.3.5.1.*	Maximum likelihood modeling
5.2.3.5.2.	Maximum entropy modeling
5.2.3.5.3.	Maximum a posteriori modeling
5.2.3.5.4.*	Mixture models
5.2.3.5.5.	Latent variable models
5.2.3.5.6.*	Bayesian network models
5.2.3.5.7.**	Markov network models
5.2.3.6.	Learning linear models
5.2.3.6.1.	Perceptron algorithm
5.2.3.6.2**	Linear discriminant analysis
5.2.3.6.2.1**	Tensor representation
5.2.3.7.*	Factorization methods
5.2.3.7.1.*	Non-negative matrix factorization
5.2.3.7.2.	Factor analysis
5.2.3.7.3.	Principal component analysis
5.2.3.7.3.1**	2D PCA
5.2.3.7.3.2**	Sparse PCA
5.2.3.7.4.	Canonical correlation analysis
5.2.3.7.5.*	Latent Dirichlet allocation
5.2.3.7.6.**	Independent component analysis
5.2.3.7.7**	Nonlinear principal components
5.2.3.7.8**	Multidimensional scaling
5.2.3.7.8.1**	Least moduli
5.2.3.8.	Rule learning
5.2.3.8.1.**	Neuro-fuzzy approach
5.2.3.9.– 5.2.3.13.	···
5.2.3.13.1.*	Deep belief networks
5.2.3.14**	Multiresolution
5.2.3.15**	Support vector machines
5.2.4.	Machine learning algorithms
5.2.4.1.	Dynamic programming for Markov decision processes
5.2.4.1.1.– 5.2.4.2.2.	···
5.2.4.2.3.**	Fusion of classifiers
5.2.4.3.	Spectral methods
5.2.4.3.1**	Spectral clustering
5.2.4.4.	Feature selection
5.2.4.5.	Regularization
5.2.4.5.1**	Generalized eigenvalue
5.2.5.	Cross-validation

Taxons that have been seriously affected by a scientist from our sample are marked with a star. Taxons added to better reflect ground-breaking results from the sample are marked with two stars

the number of successful PhDs supervised, journal editing positions, and chairing at conferences. Unfortunately, not all of the addressees replied to our messages, so we had to remove from the sample those whose merit data were missing. In our final sample there are 30 active scientists in the domain.

Now comes a most controversial part of this project—establishing which areas of the domain have been developed or transformed by this or that scientist from the sample. One of the aspects under fire is crediting somebody for this or that result. Indeed, in the current era of globalization any idea of merit can be traced back to, usually, multiple origins. We accept an easy touch position so that individuals are credited with innovations if this is what they claim themselves, and an important part of the community does support the claim. Another issue is a correct interpretation of the set of main contributions by a person. How can one select the most important items from a few hundred publications? In no way can we claim that our selections have been correct in all the cases; we only hope that did not do much harm because we selected a number of publications, usually from 4 to 6, (co)authored by each scientist from our sample. Another, even more, controversial issue is of choosing subdomains in the taxonomy drastically affected by this or that publication. This is accompanied with a bunch of more-or-less arbitrary decisions starting from deciding was this or that effect drastic indeed and finishing by a decision to add this or that node to the taxonomy. Luckily, the ACM CCS 2012 taxonomy is flexible enough to admit different interpretations of the same term. For example, "Clustering" appears in it as part of 1.1 Theory and algorithms for application domains, as well as part of 3.2. Information systems applications, as well as part of 5.2. Machine learning. This allows to properly choose a location within the taxonomy for both algorithms, systems and applications.

All in all, our main argument for the usefulness of our approach is a clear visibility of the entire argument from a piece of work (paper) to formulation of a result to mapping that to a specific (set of) node(s). This gives an opportunity to operationally discuss and correct, if needed, any part of the picture. The only issue preventing us from presenting all the detail of the data set and its mapping to the taxonomy is that the project involves scientists' names. We think that there is a kind of an implicit universal nondisclosure agreement making it inconvenient to collect a dataset about peer scientists for publicly ranking them without their consent or even their knowledge of that. The only exception from this "agreement" that can be admitted here are the names of Dr. Panos Pardalos and Dr. Boris Mirkin. There are two reasons for that. First, each of the two did want to be included into the sample. Second, this disclosure makes an evidence that our data relate to real, not imaginary, scientists. Therefore, we report here that P. Pardalos is labeled S19 and Boris Mirkin S5, in our sample.

The results of mapping of scientists from our sample to the taxonomy are presented in Table 4. The table also presents the derived taxonomic ranks and the same ranks, 0–100 normalized. The normalization went according to the accepted rule except that the minimum rank, 3.50, gets a 100 mark, and the maximum one, 4.89, gets a 0. By looking at the values of the taxonomic rank, it seems quite obvious

Table 4 Mapping main research results to the taxonomy; layers of the nodes affected; Tr, taxonomic ranks derived from them; Trn, taxonomic ranks normalized to the range 0–100; and three strata obtained by k-means partitioning of the ranks

Scientist	Mapping to taxonomy	Layers	Tr	Trn	Stratum
S1	4.1.2.7, 5.2.1.2.7, 5.2.3.7.7	4,5,5	3,88	73	1
S2	2.1.1.2, 2.1.1.2, 5.2.2.7, 5.2.3.5, 5.2.3.5	4,4,4,4,4	3,50	100	1
S3	3.2.1.4.2, 5.2.1.2.3, 5.2.1.2.7, 5.2.3.5.4, 5.2.3.7.6	5,5,5,5,5	4,50	29	2
S4	1.1.1.4.3, 3.4.4.5, 5.2.1.1.1,5.2.1.2.7, 5.2.3.2.1,5.2.3.7.8	5,4,5,5,5,5	3,90	71	1
S5	3.2.1.4.4, 3.2.1.4.4, 3.2.1.4.5, 3.2.1.4.6, 3.2.1.11.1	5,5,5,5,5	4,50	29	2
S6	3.1.1.5.2, 3.1.2.1.1, 3.1.2.1.1, 3.2.1.6., 3.2.1.7	5,5,5,4,4	3,77	81	1
S7	5.2.3.5.6, 5.2.3.5.7	5,5	4,80	7	3
S8	3.2.1.3.1, 3.2.1.4.1, 5.2.3.3.1, 5.1.3.2.1, 5.1.3.2.4	5,5,5,5,5	4,50	29	2
S9	5.2.1.2.3, 5.2.3.3.2, 5.2.3.5.1, 5.2.3.5.4, 5.2.3.6.2	5,5,5,5,5	4,50	29	2
S10	5.2.3.3.2, 5.2.3.13.1	5,5	4,80	7	3
S11	3.2.1.2, 3.2.1.2.1, 3.2.1.3.3, 3.2.1.4.1, 3.2.1.7.2	4,5,5,5,5	3,86	74	1
S12	3.2.1.9.1.1, 3.2.1.10, 3.2.1.11.2, 5.1.1.7.1, 5.2.3.1.3, 5.2.3.4.1	6,4,5,5,5,5	3,86	74	1
S13	1.1.1.3, 5.2.1.2.1, 5.2.1.2.1, 5.2.2.7.1, 5.2.3.7.1	4,5,5,5,5	3,86	74	1
S14	3.2.1.3.1	5	4,90	0	3
S15	5.2.4.3.1	5	4,90	0	3
S16	5.2.4.2.3	5	4,90	0	3
S17	2.1.3.7.1, 5.2.4.3.1, 5.2.3.7.5., 5.2.1.2.4, 5.2.3.2.4, 5.2.3.7.3.2, 5.2.3.5.4., 5.2.4.3.1	5,5,5,5,6,5,5	4,39	36	2
S18	3.2.1.9.1, 3.2.1.9.2, 5.2.3.3.3.1	5,5,6	4,79	8	3
S19	3.2.1.7.5, 3.2.1.9.3, 5.2.3.2.1.1, 5.2.4.5.1	5,5,6,5	4,69	15	3
S20	3.2.1.4.3, 5.2.3.7.7, 5.2.3.7.8.1	5,5,6	4,79	8	3
S21	1.1.1.6, 2.1.1.2, 2.1.1.8.3, 3.2.1.6, 3.4.1.6, 5.1.2.4, 5.2.1.1.3	4,4,5,4,4,4,5	3,57	95	1
S22	3.2.1.2.2, 5.2.1.2.7.1, 5.2.3.1.2, 5.2.3.6.2.1	5,6,5,6	4,78	9	3
S23	3.2.1.3, 3.2.1.3.1, 3.4.4.1	4,5,4	3,79	79	1
S24	2.1.5.3.1	5	4,90	0	3
S25	5.2.3.3.3.2, 5.2.3.8.1	6,5	4,89	1	3
S26	3.2.1.11.1, 3.2.1.11.1, 3.3.1.6, 5.2.2.7, 5.2.3.5.6	5,5,4,4,5	3,77	81	1
S27	3.2.1.3.2, 3.2.1.4.1, 5.2.1.2.1, 5.2.3.1.1	5,5,5,5	4,60	21	2
S28	3.2.1.8	4	3,90	71	1
S29	5.2.3.3.2.1, 5.2.3.3.3.3, 5.2.3.3.4	6,6,5	4,88	1	3
S30	5.1.3.2.1.1, 5.2.1.2.7.2, 5.2.3.3.5	6,6,5	4,88	1	3

that the number of strata should be set to 3, as most values concentrate around 0, 30, and 70 or more. This specifies the number of strata to look for over all the criteria under consideration.

3.3 Scoring Citation and Merit

There are a number of engines to score citation indexes of scientists. They are slightly differing over the databases of publications involved or the time periods used in evaluations or some measure modifications. Yet there are no verified claims of superiority or inferiority of ones over others. Therefore, we limit ourselves with the citation indexes routinely available at Google Scholar. The three metrics readily available for every scientist who has arranged her/his Scholar Google profile are:

- Number of citations that the scientist has received (Citations);
- Number of her/his papers that received at least 10 citations (#10);
- Hirsch index (H): The number h of papers that received at least h citations.

Table 5 contains values of the three criteria in July 2013 as well as the gain values, percent, showing how much they increased to September 2014. Three columns on the right present criteria values in 2014 normalized so that the minimum is 0 and maximum, 100. Although some empirical proof of stability of the Linstrat stratification method has been described in [23], these two data sets can be used to further test the stability of the method.

Merit of a scientist is a rather vague concept to represent the level of services to and appreciation of the scientist by the "research community." Of many possible criteria we select those related to the success of the "research school" established by the scientists and the level of recognition of them. Of course, the levels of citations reflect both. Yet here we are going to use measures related to personal efforts made and personal positions taken by a scientist. The success manifests itself both scientifically and administratively. The former can be measured by the number of successful PhD students by the scientist. The latter can be measured by the number of research publishing journals at which the scientist has a role. The level of recognition can be measured by the number of conferences at which the scientist has been invited to give a plenary presentation or to participate in organization of. Therefore, we use the following criteria of merit:

- Number of successful PhD students supervised (PDS);
- Number of scientific journals in which they have been chief or associate editor (at any time) or a member of the editorial board currently (EJ);
- Number of conferences at which they have participated as either chair or co-chair or program-chair or keynote-chair or deputy chair or global chair (CC).

These data over our sample of 30 scientists are presented in Table 6.

Table 5 Citation metric scores: total number of citations, number of papers received 10 or more citations, and Hirsch index

	In 2013			Gain (%)			Normalized		
Scientist	Citations	#10	Hirsch	Citations	#10	Hirsch	Citations	#10	Hirsch
S1	5,138	101	32	11	6	3	0	8	9
S2	37,371	175	78	15	4	4	20	20	46
S3	113,240	476	144	14	6	4	68	70	100
S4	70,932	292	98	17	15	5	41	40	63
S5	5,205	61	31	16	7	3	0	2	8
S6	47,844	316	96	15	10	8	27	44	61
S7	38,862	299	97	16	44	4	21	41	62
S8	9,400	119	46	14	7	2	3	11	20
S9	26,630	134	42	18	12	8	14	14	17
S10	92,538	239	102	32	4	15	55	31	66
S11	39,468	182	73	13	6	6	22	22	42
S12	55,831	220	65	16	4	5	32	28	36
S13	14,653	104	53	18	12	6	6	9	26
S14	95,598	608	122	19	40	7	57	91	82
S15	84,127	179	83	25	7	4	50	21	50
S16	12,028	86	45	17	10	7	4	6	20
S17	77,512	342	116	19	12	9	45	48	77
S18	30,009	150	65	14	8	7	16	16	36
S19	26,220	402	76	7	7	1	13	58	45
S20	5,408	50	21	2	6		0	0	0
S21	24,117	121	70	14	7	9	12	12	40
S22	18,665	260	70	26	12	11	9	34	40
S23	82,781	203	89	10	4	1	49	25	55
S24	164,251	280	108	16	10	7	100	38	71
S25	5,530	50	29	16	11	7	0	0	7
S26	29,334	155	65	11	8	5	15	17	36
S27	54,579	661	87	11	23	4	31	100	54
S28	54,098	472	111	1	1	0	31	69	73
S29	23,773	309	69	16	14	10	12	42	39
S30	14,954	179	61	31	20	13	6	21	33

Columns 2–4 contain values of criteria in 2013. Columns 5–7 show the gains of the corresponding metrics in 2014. Last three columns are 2014 values normalized from 0 to 100

3.4 Combined Criteria and Stratifications Obtained

Here are the Linstrat based analyses that we conducted over the data in Tables 4, 5, and 6:

1. Found a 3-strata stratification over three citation features in Table 5. The combined criterion is formed with weights 0.5, for Citations, 0.5, for #10, and 0

Table 6 Three merit criteria: PDS, number of successful PhDs supervised, CC, number of conferences (co)chaired, EJ, the number of journals (co)edited

	Merits			Normalized values		
Scientist	PDS	CC	EJ	PDS	CC	EJ
S1	28	5	2	49	6	3
S2	15	12	4	22	16	8
S3	38	24	9	69	31	22
S4	9	5	8	10	6	19
S5	16	21	4	24	27	8
S6	18	6	1	29	8	0
S7	4	0	1	0	0	0
S8	7	19	6	6	25	14
S9	11	5	16	14	6	42
S10	30	36	2	53	47	3
S11	12	7	5	16	9	11
S12	5	20	6	2	26	14
S13	8	7	5	8	9	11
S14	8	11	2	8	14	3
S15	31	3	2	55	4	3
S16	5	1	2	2	1	3
S17	34	2	8	61	3	19
S18	12	6	6	16	8	14
S19	53	77	27	100	100	72
S20	10	2	5	12	3	11
S21	9	7	1	10	9	0
S22	6	18	8	4	23	19
S23	9	9	9	10	12	22
S24	17	3	8	27	4	19
S25	7	7	3	6	9	6
S26	30	30	6	53	39	14
S27	25	28	12	43	36	31
S28	16	29	37	24	38	100
S29	13	28	15	18	36	39
S30	7	16	17	6	21	44

Columns 2 through 4 contain real counts, and columns 5–7 are those 0–100 normalized

for Hirsch over the data at 2014. For the data of 2013, the respective weights are 0.44 (Citations), 0.56 (#10), 0 (Hirsch). Given that the Citations criterion grew by two-digit percentage points from 2013 to 2014 at 90 % of the sample while the #10 criterion by only a one-digit percent value in most cases, the change of the weights between the two criteria from 2013 to 2014 is consistent. The fact that the Hirsch index criterion's weight is 0 in both cases goes in line with the overwhelming critiques the criterion has been exposed to recently, see [2, 25, 28, 33].

Table 7 Weights of individual criteria in those combined, Citation combined, Merit combined, and Research impact panoramic

Citation		Merit		Panoramic	
Citations	0.5	PDS	0.22	Taxonomic rank	0.80
#10	0.5	CC	0.10	Citation combined	0.04
Hirsch	0.0	EJ	0.69	Merit combined	0.16

2. Found a 3-strata stratification over three merit features in Table 6. The combined criterion is formed with weights 0.22 at PDS, 0.10 at CC, and 0.69 at EJ. The relative weight values are consistent with our intuition based upon the prevailing practice of maintaining a heavy and just submission reviewing process in leading journals.
3. Took the two found combined criteria, for citation and merit, and considered them together with the taxonomic rank to find a panoramic stratification embracing all the three aspects of the researcher's impact: level of results, level of citation, and level of merit. The combined panoramic criterion is formed by summing those three with the weights 0.80 (Taxonomy rank), 0.04 (Combined citation), and 0.16 (Combined merit), which also corresponds to our intuition.

For the sake of convenience, we summarize these results in Table 7, for the weights, and in Table 8, for the combined criteria and stratifications.

To summarize these results in general, let us take Pearson correlation coefficients between the four criteria, Cc, Mc, T, and P, as well as Spearman correlation coefficients between the stratification rankings, Cs, Ms, Ts, and Ps. They are presented in Table 9.

As one can see, Pearson and Spearman results are much similar. The three aspects under consideration, Citation, Merit, and Taxonomy rank, are rather uncorrelated pairwise, which justifies, up to the extent of the representativeness of our sample, the choice for measurement scales of these aspects. Yet the two indirect scales, Citation and Merit, are somewhat positively correlated, probably to that extent at which they both relate to the popularity of a scientist. Of course, the comprehensive Panoramic criterion much correlates with its major constituent, the Taxonomy rank. Especially impressive this correlation is at the stratifications: Ps almost coincides with Ts, differing from Ts by just one scientist's move from stratum 3 to stratum 2.

On the level of individual researchers, S5 and S19, their lot put them into the middle lane, stratum 2, of the Panoramic scale. Yet the trajectories are different. Scientist S5, Boris Mirkin, makes very little on both, Citation and Merit, scales, yet falls in stratum 2 over the Taxonomy. In contrast, scientist S19 is good on both Citation and Merit, especially on the latter, where he is the best of the entire sample and shares the stratum Ms = 1 with just one other researcher. He falls within Ps = 2 just because the papers that have been published by him on data analysis, although quite fine from the optimality point of view, did not pay much attention to the structure of the data analysis area. It seems rather obvious that with the publication

Table 8 Stratifications and combined criteria values at the sample of scientists over various sets of criteria: Cc and Ccn, citation combined criterion values as computed and normalized to 0–100 scale, respectively; Mc and Mcn, merit combined criterion values as computed and normalized to 0–100 scale, respectively; Trn, taxonomic rank normalized; P and Pn, panoramic combined criterion values as computed and normalized to 0–100 scale, respectively; Cs, Ms, Ts, and Ps, three-strata stratifications over criteria Ccn, Mcn, Trn, and Pn, respectively

Scientist	Cc	Ccn	Mc	Mcn	Tr	P	Pn	Cs	Ms	Ts	Ps
S1	4	5	13	17	73	61	73	3	3	1	1
S2	20	27	12	15	100	84	100	3	3	1	1
S3	69	93	33	41	29	33	39	1	2	2	2
S4	41	55	16	19	71	62	74	2	3	1	1
S5	1	1	13	17	29	26	30	3	3	2	2
S6	35	48	7	9	81	68	81	2	3	1	1
S7	31	42	0	0	7	7	8	2	3	3	3
S8	7	9	13	16	29	26	31	3	3	2	2
S9	14	19	32	40	29	30	36	3	2	2	2
S10	43	58	18	23	7	11	13	2	3	3	3
S11	22	30	12	15	74	63	75	3	3	1	1
S12	30	41	12	15	74	63	76	2	3	1	1
S13	7	10	10	13	74	62	74	3	3	1	1
S14	74	100	5	6	0	5	5	1	3	3	3
S15	36	48	15	18	0	5	5	2	3	3	3
S16	5	7	3	3	0	1	0	3	3	3	3
S17	46	63	27	33	36	37	43	2	2	2	2
S18	16	22	14	17	8	10	11	3	3	3	3
S19	35	48	81	100	15	30	35	2	1	3	2
S20	0	0	10	13	8	8	9	3	3	3	3
S21	12	16	3	4	95	78	93	3	3	1	1
S22	21	29	16	20	9	11	13	3	3	3	3
S23	37	50	18	23	79	70	83	2	3	1	1
S24	69	93	19	24	0	7	8	1	3	3	3
S25	0	0	6	8	1	2	2	3	3	3	3
S26	16	22	25	31	81	71	84	3	2	1	1
S27	65	88	34	42	21	27	32	1	2	2	2
S28	50	68	77	96	71	75	89	2	1	1	1
S29	27	36	34	42	1	9	10	2	2	3	3
S30	13	18	33	41	1	8	9	3	2	3	3

Table 9 Pairwise correlation between criteria and between stratifications

Pearson				Spearman			
Criterion	Ccn	Mcn	P	Stratification	Cs	Ms	Ps
Tr	−0.12	−0.04	0.99	Ts	−0.12	−0.02	0.98
Cc		0.31	−0.04	Cs		0.25	−0.10
Mc			0.10	Ms			0.06

of results in this volume, P. Pardalos will be getting a higher rank of the ACM CCS taxonomy, which should propel him to much higher scores on that in a very near future.

4 Conclusion

This chapter attempts at taking a more rounded view on the problem of evaluating impact of a researcher than it is assumed usually. Rather than concentrate on conventional citation scoring or more recent network related scoring or even somewhat controversial peer-review evaluations, we come up with an idea that the impact cannot be properly evaluated without looking at the meaning and level of the research results obtained by scientists. We realize that the idea is not quite novel. Yet we suggest an operational approach to implement the idea by mapping the published research results to a taxonomy of the domain and we show how this can be done by developing an example of such an evaluation. The example concerns the very area at which we conduct our research projects ourselves, the domain of data analysis, data mining, and machine learning. We take a small sample of scientists in this area so that we are able to manually map their research results to a suitable taxonomy, which is an adaptation of the ACM CCS 2012 taxonomy.

We also tackle two other dimensions of the impact, citation and merit, by taking three operationally defined criteria for each. To combine the criteria, we use another in-house idea of finding such a weighting of them which approximates the Pareto slices with between-hyperplane layers. Although rather unconventional, this approach has been found competitive in our previous work [23, 24].

Our empirical results are well matching the conventional wisdom, which may seem rather suspicious. But they all have been computed from the data without any attempt at trimming them. We make our data available so that everybody could make her/his own computations. First of all, the controversial Hirsch index to score the citation levels appears quite homely here: it gets a zero weight, so it is out of the picture by itself. Second, when combining the found scales for all three dimensions, Citation combined, Merit combined, and Taxonomic rank, the latter much outweighs the others by getting the weight of 80 % in the combined, Panoramic, criterion. Third, the three dimensions are mutually uncorrelated, except for a small positive

correlation between the Citation and Merit combined, probably because both reflect popularity of a scientist.

This suggests directions for future work. First of all, one needs to extend the empirical research both in getting larger samples and tackling on other research domains. Second, we should try automating the task of mapping one's research results to the taxonomy. Third, we should take a look whether other uncorrelated dimensions for research impact exist and, if yes, what are they and how one could measure them. Making these and similar steps will bring us closer to the final goal of developing a comprehensive measure of research impact.

Acknowledgements This work was partially supported by the International Laboratory of Decision Choice and Analysis as part of a project within the Program for Fundamental Research of the National Research University Higher School of Economics Moscow.

References

1. Abramo, G., Cicero, T., D'Angelo, C.A.: National peer-review research assessment exercises for the hard sciences can be a complete waste of money: the Italian case. Scientometrics **95**(1), 311–324 (2013)
2. Albert, B.: Impact factor distortions. Science **340**(6134), 787 (2013)
3. Aleskerov, F.T., Chistyakov, V.V., Kalyagin, V.A.: Multiple criteria threshold decision making algorithms, 40 pp. Working paper WP7/2010/02. State University Higher School of Economics, Moscow (2010)
4. Aragn, A.M.: A measure for the impact of research. Sci. Rep. **3**, 1649 (2013). doi:10.1038/srep01649
5. Bollen, J., Van de Sompel, H., Hagberg, A., Chute, R.: A principal component analysis of 39 scientific impact measures. PloS ONE **4**(6), e6022 (2009)
6. Brans, J.P., Vincke, P.: A preference ranking organisation method: the PROMETHEE method for MCDM. Manag. Sci. **31**(6), 647–656 (1985)
7. Burgess, A., Davies, U., Doyle, M., Gilbert, A, Heine, C., Howard, C, Jones, S., McKelvey, D., Potter, K., Wright, S.: The Economists Pocket World in Figures: 2007 Edition, 254 pp. The Economist in Association with Profile Books Ltd., London (2006)
8. Canavan, J., Aisling, G., Aileen, S.: Measuring research impact: developing practical and cost-effective approaches. Evid. Policy **5**(2), 167–177 (2009)
9. Choo, E.U., Bertram, S., William, C.W.: Interpretation of criteria weights in multicriteria decision making. Comput. Ind. Eng. **37**(3), 527–541 (1999)
10. Eisen, J.A., MacCallum, C.J., Neylon, C.: Expert failure: re-evaluating research assessment. PLoS Biol. **11**(10), e1001677 (2013). doi:10.1371/journal.pbio.1001677
11. Engels, T.C., Goos, P., Dexters, N., Spruyt, E.H.: Group size, h-index, and efficiency in publishing in top journals explain expert panel assessments of research group quality and productivity. Res. Eval. **22**, 224–236 (2013)
12. Figueira, J.R., Greco, S., Roy, B., Sowiski, R.: An overview of ELECTRE methods and their recent extensions. J. Multi-Criteria Decis. Anal. **20**(1–2), 61–85 (2013)
13. Fisher, W.D.: Clustering and Aggregation in Economics. The Johns Hopkins Press, Baltimore (1969)
14. Han, J., Kamber, M., Jian P.: Data Mining: Concepts and Techniques, 3rd edn. The Morgan Kaufmann Series in Data Management Systems. Morhgan Kaufmann, Amsterdam (2011)

15. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, New York (1976)
16. Kksalan, M., Mousseau, V., Ozpeynirci, O., Ozpeynirci, S.B.: An outranking-based approach for assigning alternatives to ordered classes. Nav. Res. Logist. **56**(1), 74–85 (2009)
17. Lee, F.S., Pham, X., Gu, G.: The UK research assessment exercise and the narrowing of UK economics. Camb. J. Econ. **37**(4), 693–717 (2013)
18. Mirkin, B.: Core Concepts in Data Analysis: Correlation, Summarization, Visualization. Springer, London (2011)
19. Mirkin, B.: On the notion of research impact and its measurement. Control Large Syst. Spec. Issue **44**, 292–307 (2013). Institute of Control Problems, Moscow (in Russian)
20. Mirkin, B., Orlov, M.: Methods for Multicriteria Stratification and Experimental Comparison of Them, p. 31. Preprint WP7/2013/06. Higher School of Economics, Moscow (2013, in Russian)
21. Ng, W.L.: A simple classifier for multiple criteria ABC analysis. Eur. J. Oper. Res. **177**, 344–353 (2007)
22. Nobel Prize page: http://nobelprize.org/alfred_nobel/will/will-full.html (2014). Accessed 16 Oct 2014
23. Orlov, M.: An algorithm for deriving a multicriterion stratification. Bus. Inf. **4**, 24–35 (2014, in Russian)
24. Orlov, M., Mirkin, B.: A concept of multicriteria stratification: a definition and solution. Procedia Comput. Sci. **31**, 273–280 (2014)
25. Osterloh, M., Frey, B.S.: Ranking games. Eval. Rev. (2014). doi:10.1177.0193841X14524957
26. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. Stanford InfoLab (1999)
27. Ramanathan, R.: Inventory classification with multiple criteria using weighted linear optimization. Comput. Oper. Res. **33**, 695–700 (2006)
28. San Francisco Declaration on Research Assessment (DORA): am.ascb.org/dora/ (2014). Accessed 16 Oct 2014
29. Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., Wu, T.: RankClus: integrating clustering with ranking for heterogeneous information network analysis. In: Proceedings of EDBT 2009, pp. 565–576 (2009)
30. The 2012 ACM Computing Classification System. http://www.acm.org/about/class/2012 (2014). Accessed 17 Oct 2014
31. The Complete University League Guide: http://www.thecompleteuniversityguide.co.uk/league-tables/methodology (2014). Accessed 25 Oct 2014
32. Thompson Reuters Intellectual Property and Science: http://ip-science.thomsonreuters.com/. Accessed 16 Oct 2014
33. Van Raan, A.F.: Comparison of the Hirsch-index with standard bibliometric indicators and with peer judgment for 147 chemistry research groups. Scientometrics **67**(3), 491–502 (2006)

SVM Classification of Uncertain Data Using Robust Multi-Kernel Methods

Raghav Pant and Theodore B. Trafalis

Abstract In this study we have developed a robust Support Vector Machines (SVM) scheme of classifying uncertain data. In SVM classification data uncertainty is not addressed efficiently. Furthermore, while traditional SVM methods use a single kernel for learning, multiple kernel schemes are being developed to incorporate a better understanding of all the data features. We combine the multiple kernel learning methods with the robust optimization concepts to formulate the SVM classification problem as a semi-definite programming (SDP) problem and develop its robust counterparts under bounded data uncertainties. We present some preliminary experimental results with some known datasets by introducing noise in the data. Initial analysis shows the robust SDP-SVM model improves classification accuracy for uncertain data.

Keywords Classification • Support vector machines • Multiple kernel learning • Robust optimization

1 Introduction

Over the years data mining algorithms have become popular due to their ability to find patterns and make predictions for large and complex data sets. Machine learning techniques such as support vector machines (SVMs) [13] provide good classifiers for most type of data. In many cases the data sets contain uncertain or noisy data. The term noisy data is typically meant to represent those samples that do not lie on the intended side of the separation margin. In this work, we extend

R. Pant
Environmental Change Institute, University of Oxford, South Parks Road, Oxford OX1 3QY, UK
e-mail: raghav.pant@ouce.ox.ac.uk

T.B. Trafalis (✉)
School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK 73019, USA
e-mail: ttrafalis@ou.edu

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_13

the term to include the uncertainty that manifests itself in every data point. Hence, our interpretation of noisy data consists of both (a) an error in measurement of each data point that must be considered during classification and (b) situations where data points are classified incorrectly.

Traditional SVM methods adjust for the misclassification error relative to the maximum margin of classification, but do not consider individual data uncertainties, as they are based on deterministic formulations. Bhattacharyya et al. [2] addressed such issues by developing second order conic programming (SOCP) SVM formulations for Gaussian uncertainty in data, which bore resemblance to the Total SVM methods of [3] that provided SVM formulations for bounded uncertainties. While these methods were developed separately they fall under the scheme of robust SVM approaches detailed in works of Trafalis et al. [10, 11], which use concepts developed in robust optimization (RO) literature [1]. The SOCP robust SVM approach is a particular case of the broader semi-definite programming (SDP) [12] approach for solving such problems.

Generally in the traditional SVM approach the input data reside in a kernel matrix, which is fixed when the optimization problem is solved. The kernel function is generally tuned via bootstrapping techniques to infer the best kernel matrix for classification. Using one kernel matrix is sometimes disadvantageous because there could be multiple patterns of data, which are not best represented through a single kernel function. Lanckriet et al. [8] have suggested that using a weighted linear combination of different kernel matrices can improve the SVM classification problem. Moreover when using multiple kernel matrices we do not need to tune their parameters, but instead the weights in the linear combination can be made as decision variables in the SVM classification problem. Such kernel-based learning methods of the SVM classification generally are formulated as SDP models [8], which is explained later in the chapter. Also the penalty parameter for misclassification errors is generally assumed priori before solving the SVM optimization problem. In the kernel-based learning SDP model this misclassification penalty term is also incorporated as a decision variable in the optimization problem, which means we solve for misclassification via theoretical methods rather than the traditional bootstrapping techniques.

The main contribution of the chapter is the theoretical development of a computationally tractable robust SDP-SVM formulation of the nominal multiple kernel learning SDP formulation derived from the 2-norm soft margin SVM classification problem, when we have data points with bounded uncertainties. While we deal specifically with Euclidean uncertainty sets, our formulation applies to any norm-bounded data uncertainty set. We have also addressed the issue of mapping uncertainties from the input space to the feature space, which is important for developing robust approaches for bounded uncertainty sets. While the problem size of the robust SDP is considerably increased, leading to computational issues, it is able to produce satisfactory classification results.

This chapter is organized in the following order. In Sect. 2 we provide some mathematical notation used in the rest of the chapter. Section 3 introduces the SVM learning problem, and Sect. 4 discusses our SDP formulation that makes the

problem dependent upon learning of the kernel matrix and the penalty parameter. In Sect. 5 we present the concept of bounded uncertainty in the input space and its extension into the feature space. We propose the robust extensions to the SDP learning problem and show the proof for getting such formulations. Finally, we present the robust SDP problem for SVM classification. In Sect. 6 we show how our methods are applicable for data sets with uncertainties and present some initial computational results of our scheme for benchmark data sets, with concluding remarks given in Sect. 7.

2 Notation

Vectors are represented in bold, e.g., $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{e}_m$ is an $m \times 1$ vector of ones, while scalars are in lower case italics, e.g., z. Matrices are represented in upper case bold, e.g., $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\mathbf{I}_m$ is used for denoting the $m \times m$ identity matrix. For a vector the relation $\mathbf{a} \geq 0$ means that each element of the vector is non-negative, while for a matrix the relation $\mathbf{X} \succeq 0$ implies that the matrix is positive semi-definite. The dot product between two vectors $\mathbf{a}$ and $\mathbf{b}$ is represented either as $\mathbf{a}^T\mathbf{b}$ or as $\langle \mathbf{a}, \mathbf{b} \rangle$.

3 Two-Norm Support Vector Classification

In a two-class classification problem we assume we are given a set of training data $\{\mathbf{x}_i, y_i\}_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^n$ represents an input data sample and $y_i \in \{-1, +1\}$ represents its assigned class (output). The aim of the SVM classification problem is to find the best hyperplane that separates the two classes of data. If the input data are linearly separable then this hyperplane is of the form $\langle \mathbf{w}, \mathbf{x} \rangle + b$. In most real-world problems linear separation is not possible, so to address the problem of non-linearity we map the input data points $\{\mathbf{x}_i\}_{i=1}^m$ to a higher dimensional feature space $\mathcal{F}$ where the data are linearly separable. If we are able to compute a function $\phi : \mathbf{x} \to \phi(\mathbf{x}) \in \mathcal{F}$ to map the input co-ordinates onto the feature space, our hyperplane is now of the form $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b$. In SVM classification the hyperplane that divides the two classes with maximum distance is the best hyperplane. Mathematically this hyperplane $(\mathbf{w}^*, b^*)$ is obtained by solving the optimization problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geq 1, \quad (i = 1, 2, \ldots, m), \end{aligned} \tag{1}$$

producing the maximum margin ($\gamma = 1/||\mathbf{w}_*||_2$) of separation between the classes of data. Solution (1) is generally referred to as a hard margin solution, which exists when the data are perfectly linearly separable in the feature space. Since this mostly

is not the case we generally solve a soft margin SVM classification problem for non-separable data. In this chapter we take the 2-norm soft margin SVM classification to find the maximal margin hyperplane $(\mathbf{w}^*, b^*)$ that separates the data, via the optimization problem

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \ \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + \frac{1}{2\tau}\langle \xi,\xi\rangle \\ &\text{subject to} \ \ y_i(\langle \mathbf{w},\phi(\mathbf{x}_i)\rangle + b) \geq 1-\xi_i, \quad (i=1,2,\ldots,m), \end{aligned} \tag{2}$$

Here in (2) $\{\xi_i\}_{i=1}^m$ are slack variables that signify misclassification errors for each data point $\phi(\mathbf{x}_i)$, and $\tau > 0$ is the penalty parameter on these misclassification errors. Smaller values of τ indicate that the width of the margin of separation ($\gamma = 1/||\mathbf{w}_*||_2$) is smaller as more weight is given to correctly classified points near the separation boundary. As τ increases the separation margin increases as farther points are also used as support vectors to the separating hyperplane [7].

In most cases it is impossible to compute the function $\phi(\mathbf{x}_i)$ but it is easier to find the inner product between pair of co-ordinates in the feature space [4]. For two points $\mathbf{x}_i$ and $\mathbf{x}_j$, the inner product between the mappings $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ in the $\mathcal{F}$ space is called the kernel function, $k(\mathbf{x}_i,\mathbf{x}_j) = \langle \phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle$. The matrix $\mathbf{K} \in \mathbb{R}^{m\times m}$ is the kernel matrix, in which each element is given by $k_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$. Since the kernel matrix specifies the inner product between all possible pairs of training vectors, it is a measure of the relative positions of these points in the feature space [4].

To utilize the kernel matrix instead of solving for the optimization problem given by (2) we look at its Lagrangian dual formulation as given by (3):

$$g(\tau,\mathbf{K}) = \max_{\alpha}\left[\alpha^T\mathbf{e}_m - \frac{1}{2}\alpha^T(\mathbf{H}(\mathbf{K}) + \tau\mathbf{I}_m)\alpha\right] : \alpha^T\mathbf{y} = 0, \alpha \geq 0, \tag{3}$$

where $\mathbf{H}(\mathbf{K})$ is the matrix that contains the data and labels and is defined as $\mathbf{H}(\mathbf{K}) = \left[h_{ij}\right] = \left[y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)\right]$, showing its dependence on the kernel matrix. $\mathbf{y} \in \mathbb{R}^m$ is the vector of class labels $\{y_i\}_{i=1}^m$.

Given values for τ and $\mathbf{K}$ we can obtain the optimal α^* as the solution to the SVM classification problem. Those elements of α^* that are non-zero correspond to the support vectors for classification. The kernel-based SVM classification problem of (3) is convenient to compute because it is a quadratic optimization problem with linear constraints. From the Karush–Kuhn–Tucker optimality relations between the primal in (2) and the dual in (3) we obtain $\mathbf{w}^* = \sum_{i=1}^m \alpha_i^* y_i \phi(\mathbf{x}_i)$ and $b^* = \frac{1}{M}\sum_{j=1}^M \left(y_j - \sum_{i=1}^m \alpha_i^* y_i k(\mathbf{x}_i,\mathbf{x}_j)\right)$. Here we have computed b^* by averaging over the number of elements M of α^* that are non-zero, since each one of them is a support vector. For classification of any new unlabelled data point $\mathbf{x}$ we find the value of the

function $f(\mathbf{x}) = \sum_{i=1}^{m} y_i \alpha_i^* k(\mathbf{x}_i, \mathbf{x}) + b^*$ to infer the class of $\mathbf{x}$. In binary classification with ± 1 labels the decision rule defined by $\text{sign}(f(\mathbf{x}))$ is used to classify the point $\mathbf{x}$ into one of the two classes. If $f(\mathbf{x})$ is positive then $\mathbf{x}$ belongs to the class labelled as $+1$, while a negative $f(\mathbf{x})$ results in $\mathbf{x}$ belonging to the class labelled -1.

4 Kernel Learning

The solution of the classification problem specified by (3) is dependent upon $\mathbf{H}(\mathbf{K})$, and hence $\mathbf{K}$ as these matrices contain the information about the training samples. In a standard classification problem the data points $\{\mathbf{x}_i\}_{i=1}^m$ are assumed to be deterministic and as such fixed $\mathbf{K}$ values give a particular optimal separating hyperplane. The performance of the classification is improved by tuning $\mathbf{K}$ parameters through cross-validation schemes. Such a solution approach is therefore making use of $\mathbf{K}$ as a decision variable in optimization, which allows us to look at the classification problem (3) in terms of $\mathbf{K}$. In addition, the penalty parameter τ influences the misclassification and separation margin. It is also tuned through cross-validation schemes, which can lead to approximations that do not control the errors effectively. We can improve the estimation of τ by incorporating it as a decision variable in the optimization problem. We look at $g(\tau, \mathbf{K})$ from (3) and make the observation that

$$g(\tau, \mathbf{K}) = \begin{cases} \alpha^T \mathbf{e}_m \text{ if } & \frac{1}{2}\alpha^T(\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m)\alpha \geq 0 \\ +\infty & \text{otherwise} \end{cases}. \tag{4}$$

Equation (4) shows that $g(\tau, \mathbf{K})$ has a finite upper bound and feasible solution only when $\alpha^T(\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m)\alpha \geq 0$, otherwise the system becomes infeasible. Also, it can be seen that $\mathbf{H}^T(\mathbf{K}) = \left[h_{ji}\right] = \left[y_j y_i k(\mathbf{x}_j, \mathbf{x}_i)\right] = \left[y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)\right] = \mathbf{H}(\mathbf{K})$, which proves that $\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m \succeq 0$. Thus, the semi-definiteness of the matrix is important for a barrier on the feasible solution to the optimization problem.

It can also be observed that the semi-definite matrix $\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m$ will affect the optimal value of the objective $g(\tau, \mathbf{K})$. If τ grows then it can drive $g(\tau, \mathbf{K})$ towards a very low threshold. Hence, as is done in a SDP approach, we choose to have a $\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m$ matrix with a bounded trace ($r > 0$). Imposing such a condition makes sure that the optimum is bounded and also computationally the problem is easier to solve.

Function $g(\tau, \mathbf{K})$ is the maximum of affine functions of both τ and $\mathbf{K}$ and hence convex. Also the set of constraints $\alpha^T \mathbf{y} = 0, \alpha \geq 0$ is convex. If the kernel matrix belongs to a convex set $\mathcal{K}$ then we can obtain different realizations for $g(\tau, \mathbf{K})$ for varying kernel matrices in $\mathcal{K}$. Moreover, we can use a combination of kernels to improve the learning instead of using a single kernel. There are multiple patterns in data, which are not best represented through a single kernel function. A linear

combination of different kernel matrices has been shown useful in improving pattern recognition and learning [8]. Assuming $l = 1, 2, , d$ different kernel matrices each denoted by $\mathbf{K}_l$, the matrix $\mathbf{K}$ can be represented as a linear combination of different kernels as $\mathbf{K} = \sum_{l=1}^{d} \beta_l \mathbf{K}_l$, for some constants $\beta_l \geq 0$. The linear combination preserves the convexity of the problem.

Using the concepts developed so far, the best separating hyperplane for classification is the one that will solve the SDP problem

$$\begin{aligned} \min_{\tau \geq 0, \mathbf{K} \in \mathcal{K}} \quad & g(\tau, \mathbf{K}) \\ \text{subject to} \quad & \mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m \succeq 0, \\ & \text{trace}(\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m) \leq r, \\ & \mathbf{K} = \sum_{l=1}^{d} \beta_l \mathbf{K}_l, \forall \beta_l \geq 0. \end{aligned} \tag{5}$$

Lanckriet et al. [8] present a similar approach for solving the problem of obtaining labels for new data points when some of the labels are missing in classification. One of the major advantages of the above approach lies in the fact that instead of cross-validation techniques, τ is obtained through optimization techniques. Also, solutions depend upon the structures of the $\mathbf{K}$ matrix, which being positive semi-definite leads to SDP approaches to solve (5).

5 Robust SDP Formulation for Classification problem

When the available data set contains uncertain data points, the SDP in (5) becomes stochastic. For multiple linear combinations of $\mathbf{K}_l$ matrices leading to $\mathbf{K}$ we can obtain an ensemble of separating hyperplanes. Our interest lies in finding the maximal separation margin under the worst case uncertainty, which would give us a robust classification solution. In order to investigate data uncertainty in the feature space we look at the datapoint-wise worst case uncertainty in the input space and then transform it onto the feature space. The process of obtaining the robust SDP formulations follows in the subsections below.

5.1 Uncertainty Mapping for Input to Feature Space

It is assumed that every data point ($\mathbf{x}_i$) can be represented as the sum of a nominal value ($\hat{\mathbf{x}}_i$) that is free of any uncertainties and a perturbation ($\Delta \mathbf{x}_i$), which contains information about the nature of the data uncertainty. It is important to consider a convex bound for the data uncertainty because we want to preserve the convexity of the SVM classification formulations built so far. A common convex uncertainty

set for data perturbation is the spherical uncertainty set in which it is assumed that each data perturbation is contained with a sphere whose radius controls the amount of deviation from its nominal value. Hence, the data points $\{\mathbf{x}_i\}_{i=1}^m$ in our analysis belong to a convex uncertainty set $\mathcal{X}$, defined as

$$\mathcal{X} = \{\mathbf{x}_i : \mathbf{x}_i = \hat{\mathbf{x}}_i + \Delta\mathbf{x}_i, ||\Delta\mathbf{x}_i||_2 \leq \rho_i\}, \tag{6}$$

where ρ_i is the radius of the spherical uncertainty.

The transformation of the uncertainty from the input space to the feature space can become complicated because of the feature mapping. For most cases we know the kernel function and not the feature mapping; hence, such estimations are not always possible and can become more complex and dimensions increase. Since we are interested in the kernel function instead of the feature mapping, we look at the first order truncated Taylor series expansion of the kernel function in terms of the data perturbations. This is given as

$$\begin{aligned} k(\hat{\mathbf{x}}_i + \Delta\mathbf{x}_i, \hat{\mathbf{x}}_j + \Delta\mathbf{x}_j) &= \left\langle \phi(\hat{\mathbf{x}}_i + \Delta\mathbf{x}_i), \phi(\hat{\mathbf{x}}_j + \Delta\mathbf{x}_j) \right\rangle \\ &\simeq k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) + \Delta\mathbf{x}_i^T k'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) + \Delta\mathbf{x}_j^T k'_{\mathbf{x}_j}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) \end{aligned} \tag{7}$$

where $k'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)$ is the gradient of the kernel function with respect to $\mathbf{x}_i$ evaluated at the point $(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)$. Using (7) formulation $\mathbf{K}$ is divided into a nominal part $\hat{\mathbf{K}}$ and a perturbed part $\Delta\mathbf{K}$. This affects the $\mathbf{H}$ matrix also and it is expressed as $\mathbf{H} = \hat{\mathbf{H}} + \Delta\mathbf{H}$. We rewrite the $\Delta\mathbf{H}$ matrix as follows

$$\begin{aligned} \Delta\mathbf{H} = &\left[\mathbf{diag}(y_i^T)\right]\left(\left[\mathbf{diag}(\Delta\mathbf{x}_i^T)\right]\begin{bmatrix} k'_{\mathbf{x}_1}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_1) & \cdots & k'_{\mathbf{x}_1}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_m) \\ \vdots & \ddots & \vdots \\ k'_{\mathbf{x}_m}(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_1) & \cdots & k'_{\mathbf{x}_m}(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_m) \end{bmatrix}\right. \\ &\left. + \begin{bmatrix} k'^T_{\mathbf{x}_1}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_1) & \cdots & k'^T_{\mathbf{x}_m}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_m) \\ \vdots & \ddots & \vdots \\ k'^T_{\mathbf{x}_1}(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_1) & \cdots & k'^T_{\mathbf{x}_m}(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_m) \end{bmatrix}\left[\mathbf{diag}(\Delta\mathbf{x}_i)\right]\right)\left[\mathbf{diag}(y_i)\right] \\ = &\mathbf{Y}^T\Theta^T\mathbf{R}\mathbf{Y} + \mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y}. \end{aligned} \tag{8}$$

We observe that in (8) $\mathbf{Y}$ is an $m \times m$ diagonal matrix of labels, Θ is a $mn \times m$ matrix of data perturbations and $\mathbf{R}$ is a $mn \times m$ matrix of the gradient of the kernel functions with respect to the data points. Using the fact that the ith data point has a spherical bound of ρ_i, the ith column of matrix Θ will have an Euclidean norm bound given as $||\theta_i||_2 \leq \rho_i$. Instead of defining the uncertainty set $\mathcal{K}$ for the $\mathbf{K}$ matrix we can thus define an uncertainty set, $\mathcal{H}$, for the realizations of the $\mathbf{H}$ matrix as

$$\mathcal{H} = \left\{[h_{ij}] : [h_{ij}] = [\hat{h}_{ij}] + \mathbf{Y}^T\Theta^T\mathbf{R}\mathbf{Y} + \mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y}, ||\theta_i||_2 \leq \rho_i\right\}. \tag{9}$$

5.2 Robust Counterparts to Uncertain Constraints

Using the uncertainty set of (9) for the feature space, in this subsection we derive expressions for the robust SDP formulations of (5), for which we need to construct the robust counterparts of the semi-definite constraints of (5). Before stating our theorem and its proof, we need to state the *S*-Lemma, used for relating quadratic inequalities, as it is relevant to our derivations.

Definition 1. For two symmetric matrices $\mathbf{A}$ and $\mathbf{B}$ of same size, the quadratic expressions $\mathbf{z}^T\mathbf{A}\mathbf{z}+2\mathbf{a}^T\mathbf{z}+c \geq 0 \implies \mathbf{z}^T\mathbf{B}\mathbf{z}+2\mathbf{b}^T\mathbf{z}+d \geq 0$ hold true if and only if $\exists\lambda \geq 0$, such that

$$\left[\begin{array}{c|c} \mathbf{B}-\lambda\mathbf{A} & \mathbf{b}-\lambda\mathbf{a} \\ \hline \mathbf{b}^T-\lambda\mathbf{a}^T & d-\lambda c \end{array}\right] \succeq 0.$$

Theorem 1. *The Robust Counterpart*

$$\hat{\mathbf{H}}+\mathbf{Y}^T\Theta^T\mathbf{R}\mathbf{Y}+\mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y}+\tau\mathbf{I}_m \succeq 0 \quad \forall(\Theta \in \mathbb{R}^{m\times mn} : ||\theta_i||_2 \leq \rho_i) \tag{10}$$

of the matrix equation $\mathbf{H}(\mathbf{K})+\tau\mathbf{I}_m \succeq 0$ *can be equivalently represented as*

$$\left[\begin{array}{c|c} \hat{\mathbf{H}}+\tau\mathbf{I}_m-\lambda\mathbf{Y}^T\mathbf{R}^T\boldsymbol{RY} & \boldsymbol{diag}(\rho_i)\mathbf{Y} \\ \hline \boldsymbol{diag}(\rho_i)\mathbf{Y} & \lambda\mathbf{I}_m \end{array}\right] \succeq 0$$

for some $\lambda \geq 0$

Proof. Since (10) is semi-definite we can choose a non-zero real-valued vector $\xi \in \mathbb{R}^m$ for which the following conditions are equivalent

$$\begin{aligned}
&\iff \xi^T\left[\hat{\mathbf{H}}+\mathbf{Y}^T\Theta^T\mathbf{R}\mathbf{Y}+\mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y}+\tau\mathbf{I}_m\right]\xi \geq 0 \quad \forall(\xi,\Theta : ||\theta_i||_2 \leq \rho_i) \\
&\iff \xi^T\left[\hat{\mathbf{H}}+\tau\mathbf{I}_m\right]\xi+2\xi^T\mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y}\xi \geq 0 \quad \forall(\xi,\Theta : ||\theta_i||_2 \leq \rho_i) \\
&\iff \xi^T\left[\hat{\mathbf{H}}+\tau\mathbf{I}_m\right]\xi+2\min_{||\theta_i||_2\leq\rho_i}\xi^T\mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y}\xi \geq 0 \quad \forall(\xi) \\
&\iff \xi^T\left[\hat{\mathbf{H}}+\tau\mathbf{I}_m\right]\xi-2||\mathbf{diag}(\rho_i)||_2||\xi^T\mathbf{Y}^T\mathbf{R}^T||_2||\mathbf{Y}\xi||_2 \geq 0 \quad \forall(\xi) \\
&\qquad \text{[from Cauchy-Schwarz inequality]} \\
&\iff \xi^T\left[\hat{\mathbf{H}}+\tau\mathbf{I}_m\right]\xi-2||\mathbf{diag}(\rho_i)||_2||\mathbf{R}\mathbf{Y}\xi||_2||\mathbf{Y}\xi||_2 \geq 0 \quad \forall(\xi) \\
&\iff \xi^T\left[\hat{\mathbf{H}}+\tau\mathbf{I}_m\right]\xi+2\eta^T\mathbf{diag}(\rho_i)\mathbf{Y}\xi \geq 0 \\
&\qquad \forall(\xi,\eta \in \mathbb{R}^{m\times 1} : \eta^T\eta \leq \xi^T\mathbf{Y}^T\mathbf{R}^T\mathbf{R}\mathbf{Y}\xi) \\
&\iff \exists\lambda \geq 0 : \left[\begin{array}{c|c} \hat{\mathbf{H}}+\tau\mathbf{I}_m & \mathbf{diag}(\rho_i)\mathbf{Y} \\ \hline \mathbf{diag}(\rho_i)\mathbf{Y} & \end{array}\right] \succeq \lambda\left[\begin{array}{c|c} \mathbf{Y}^T\mathbf{R}^T\mathbf{R}\mathbf{Y} & \\ \hline & -\mathbf{I}_m \end{array}\right] \\
&\iff \exists\lambda \geq 0 : \left[\begin{array}{c|c} \hat{\mathbf{H}}+\tau\mathbf{I}_m-\lambda\mathbf{Y}^T\mathbf{R}^T\mathbf{R}\mathbf{Y} & \mathbf{diag}(\rho_i)\mathbf{Y} \\ \hline \mathbf{diag}(\rho_i)\mathbf{Y} & \lambda\mathbf{I}_m \end{array}\right] \succeq 0 \quad \text{[Definition 1]}
\end{aligned}$$

□

We also need to find the bound for the robust constraint, $\text{trace}(\hat{\mathbf{H}} + \mathbf{Y}^T\Theta^T\mathbf{R}\mathbf{Y} + \mathbf{Y}^T\mathbf{R}^T\Theta\mathbf{Y} + \tau\mathbf{I}_m) \leq r$ in (5). Applying the Cauchy–Schwarz inequality, we observe the following:

$$\begin{aligned}\text{trace}(\mathbf{Y}^T\Theta^T\mathbf{R}\mathbf{Y}) &= \sum_{i=1}^{m} \Delta\mathbf{x}_i^T k'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_i) \\ &\leq \sum_{i=1}^{m} \rho_i ||k'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_i)||_2 = \text{trace}(\mathbf{diag}(p_i)\mathbf{D}), \end{aligned} \tag{11}$$

where $\mathbf{D}(\in \mathbb{R}^{m\times m})$ is given as $\mathbf{D} = \left[d_{ij}\right] = \left[||k'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)||_2\right]$. Hence, in (9) the robust counterpart for the bounds on the matrix trace becomes

$$\text{trace}(\hat{\mathbf{H}} + \tau\mathbf{I}_m) + 2\text{trace}(\mathbf{diag}(p_i)\mathbf{D}) \leq r. \tag{12}$$

5.3 Robust SDP Formulation

We collect the semi-definite robust inequalities derived in Theorem 1 and (12), to present the robust SDP formulation for the 2-norm soft margin SVM classification problem. Our SDP constraints are expressed as Linear Matrix Inequalities (LMIs), due to the linear nature of the kernel matrix. This preserves convexity and provides computational solvability. The final robust SDP problem is

$$\begin{aligned} &\min_{\tau\geq 0, \mathbf{K}\in\mathcal{K}} \quad g(\tau, \mathbf{K}) \\ &\text{subject to} \quad \left[\begin{array}{c|c} \sum_{l=1}^{d} \beta_l \hat{\mathbf{H}}(\hat{\mathbf{K}}) + \tau\mathbf{I}_m - \sum_{l=1}^{d} \lambda_l \mathbf{Y}^T\mathbf{R}^T\mathbf{R}\mathbf{Y} & \mathbf{diag}(\rho_i)\mathbf{Y} \\ \hline \mathbf{diag}(\rho_i)\mathbf{Y} & \sum_{l=1}^{d} \lambda_l \mathbf{I}_m \end{array}\right] \succeq 0, \\ &\qquad \text{trace}(\sum_{l=1}^{d} \beta_l \hat{\mathbf{H}}(\hat{\mathbf{K}}) + \tau\mathbf{I}_m) + 2\text{trace}(\mathbf{diag}(\rho_i) \sum_{l=1}^{d} \beta_l \mathbf{D}) \leq r. \end{aligned} \tag{13}$$

6 Empirical Results

We choose three benchmark data sets called the *iris* [5], *breast cancer* [14], and *ionosphere* [9] available in the UCI repository. The iris data set contains 150 data samples consisting of 50 samples from three species of Iris (Iris Setosa, Iris Virginica, and Iris Versicolor). Here we assign the 50 samples of the species Iris Setosa to the minority class +1 class and the rest 100 (Iris Virginica and Iris

Versicolor) to the majority class -1. The breast cancer data set contains 239 data samples belonging to the minority class $+1$ (malignant cancer) and 444 samples in the majority class -1 (benign cancer). For the ionosphere data set 125 samples belong to the minority class $+1$ (bad) and 226 samples belong to the majority class -1 (good).

Our aim is to compare how the robust SDP-SVM model in (13) performs in comparison to the 2-norm SVM model of (2). For brevity, from this point onwards in the discussion we refer to the solution of (13) as "rSDP-SVM", while we refer the solution of (2) as "CSVM." All model analysis was run using the software CVX [6], which is a MATLAB-based modelling system developed for disciplined convex programming. For solving SDP models CVX uses the general-purpose optimization solver SeDuMi. All analysis is run on an Apple Mac computer having a 2.9 GHz processor and a 8 GB Memory.

To solve our models we first decide on the choice of kernel functions. The iris data set has been shown to be linearly separable [14], so we use the linear kernel, $k_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, for classification of the Iris data set. For the breast cancer and ionosphere data sets we use the radial basis kernel function, $k_{ij} = \exp(-0.5||\mathbf{x}_i - \mathbf{x}_j||_2^2/\sigma^2)$.

We first solve the CSVM (2) model to tune the values for the kernel parameters without any data uncertainties. These tuned parameters will be used for the rest of the analysis. For the CSVM method we fix the value of $\tau = 0.01$ for all three data sets. A tenfold cross-validation for the CSVM (2) model results in $\sigma = 84.5$ for the breast cancer data set and $\sigma = 10$ for the ionosphere data set. In the rSDP-SVM (13) model we are not concerned with tuning the kernel parameters as τ, β_l, λ_l found through the optimization adjust for kernel tuning. Computationally the rSDP-SVM (13) model has an $O(m^2 n^{2.5})$ polynomial time complexity [8], which makes it difficult to solve these problems. Our aim here in this initial analysis is to show that the model can work. Hence, at present we have taken one kernel matrix instead of multiple matrices, and hence we need to only find τ for the model. We chose the same kernel matrices given by the CSVM analysis and use these in the rSDP-SVM model. Since, tuning the kernel parameters is not required, we randomly choose data samples for training and testing. For the three data sets the training-testing sample sizes are, respectively, 70 and 30 % of entire data. We found that the rSDP-SVM gives $\tau = 2.0$ for all three data sets.

To demonstrate the usefulness of the robust SDP-SVM approach, uncertainty in these benchmark data sets is manufactured. As discussed previously, the level of uncertainty added to the data sets is determined by taking the Euclidean norm of the nominal data points and making the radius of uncertainty to be a percentage of those values. Hence, a 10 % perturbation level means that for each data point the maximum radius of the sphere of uncertainty around the nominal data point is 0.1 times the 2-norm of that nominal value. The 2-norm of the actual level of perturbation added to each data point is always kept below the radius of uncertainty. We choose 9 levels of uncertainty called *pert* levels, by incrementing the uncertainty from 0 % in steps of 5 % up to levels of 40 %. For each level of uncertainty, we

Table 1 Maximum test set accuracy for each level of uncertainty in data

	Iris		Breast cancer		Ionosphere	
Pert.(%)	CSVM	rSDP-SVM	CSVM	rSDP-SVM	CSVM	rSDP-SVM
0	100	100	98.54	**99.51**	91.89	**94.59**
5	100	100	99.03	99.03	91.89	**94.59**
10	100	100	99.03	99.03	91.89	**94.59**
15	100	100	99.03	99.03	94.59	94.59
20	100	100	98.54	**99.03**	91.89	**94.59**
25	100	100	98.54	**99.03**	91.89	**94.59**
30	100	100	98.54	**99.51**	94.59	**97.2**
35	100	100	99.03	99.03	91.89	**97.3**
40	100	100	98.54	**99.03**	91.89	**97.3**

Table 2 Number of support vectors and computation time for CSVM and rSDP-SVM simulations for each data set

		#SV	Time (s)
Iris	CSVM	3.53 ± 0.68	0.29 ± 0.06
	rSDP-SVM	12.96 ± 11.15	3.85 ± 0.36
Breast cancer	CSVM	177.04 ± 22.99	3.76 ± 0.25
	rSDP-SVM	478	76.78 ± 6.04
Ionosphere	CSVM	84.56 ± 8.11	1.44 ± 0.08
	rSDP-SVM	304.31 ± 3.48	32.71 ± 2.17

run 60 simulations again with 70–30 % training-testing division. In the rSDP-SVM procedure the value of the trace r is calculated from the given data to be equal to $\text{trace}(\mathbf{H}(\mathbf{K}) + \tau \mathbf{I}_m)$.

Two initial computational results obtained at present are discussed here. The best performances of the two methods, for each level of uncertainty, are summarized in Table 1. From Table 1 we note that the best test accuracy of both methods is same in case of the Iris dataset, but the rSDP-SVM method performs same or better than the CSVM method for the Breast Cancer and Ionosphere datasets. The cases where the rSDP-SVM method performs better are emphasized in bold in Table 1.

Table 2 shows the mean and standard deviation for the number of support vectors, across all simulations, given by the two methods for each data set. The rSDP-SVM has more support vectors than the CSVM, and it is shown in the second column of Table 2. In the rSDP-SVM method the support vectors contribute equally towards the separation margin, which also gives wider margins compared to the CSVM, where only a few points are support vectors.

Table 2 also shows the simulation time taken for each method. The rSDP-SVM is a larger problem to solve than the CSVM; hence, it is considerably slower than the CSVM. This is one of the drawbacks of using SDP methods as the computational complexity of the problem increases significantly with problem size.

We have also found out that problem sizes are limited when using CVX, which makes analysis of larger data sets an issue. Further development in algorithms and powerful computational tools would surely help in solving such issues.

7 Conclusion

The robust SDP framework developed in this chapter improves the capability of the 2-norm soft margin SVM classification. This work contributes to the development of SDP approaches to enhance pattern recognition and learning. Solving the SVM classification problem using an SDP approach helps us in better learning of the kernel matrix and also provides a theoretical justification to estimate the misclassification penalty. Data uncertainties affect the classification results and we address this problem by constructing robust counterparts of the SDP. Our robust SDP formulations involve solving LMIs which makes them computationally tractable. Hence, the main contribution of this chapter is to present the theory for deriving a computational tractable robust SDP-SVM model.

Initial computational results show that the robust SDP-SVM performs as well as, and in some cases better than, the 2-norm SVM method. The testing accuracy of the robust SDP method is high and performs better than the 2-norm SVM method. Hence, our method is capable of improving upon existing methods.

At present the drawback of the method is the increased computational complexity and large size of the problems. Due to this in the current analysis we have been limited to computing the robust SDP-SVM problem for a single kernel matrix rather that multiple kernels. Further development of this work will concentrate on improving this aspect of the analysis. Moreover the models developed here will be tested on several other data sets using several kernel matrices.

Acknowledgements The authors would like to thank the reviewers for their valuable inputs and suggested edits. The work of the author Raghav Pant was funded by the Engineering and Physical Sciences Research Council, UK, under Programme Grant EP/I01344X/1. The work of the author Theodore B. Trafalis was conducted at National Research University Higher School of Economics and supported by RSF grant 14-41-00039.

References

1. Ben-Tal, A., Ghaoui, L.E., Nemirovski, A.S.: Robust Optimization. Princeton Series in Applied Mathematics. Princeton University Press, Princeton (2009)
2. Bhattacharyya, C., Shivaswamy, P.K., Smola, A.J.: A second order cone programming formulation for classifying missing data. In: Proceedings of Neural Information Processing Systems (NIPS04) (2004)
3. Bi, J., Zhang, T.: Support vector classification with input data uncertainty. In: Proceedings of Neural Information Processing Systems (NIPS04) (2004)

4. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods. Cambridge University Press, New York (2000)
5. Fisher, R.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**, 179–188 (1936)
6. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, Version 1.21. http://cvxr.com/cvx/ (2010)
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Berlin (2003)
8. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semi-definite programming. J. Mach. Learn. Res. **5**, 27–72 (2004)
9. Sigillito, V., Wing, S., Hutton, L., Baker, K.: Classification of radar returns from the ionosphere using neural networks. J. Hopkins APL Tech. Dig. **10**, 262–266 (1989)
10. Trafalis, T.B., Alwazzi, S.A.: Support vector regression with noisy data: a second order cone programming approach. Int. J. Gen. Syst. **36**, 237–250 (2007)
11. Trafalis, T.B., Gilbert, R.C.: Robust support vector machines for classification and computational issues. Optim. Methods Softw. **22**, 187–198 (2007)
12. Vandenberghe, L., Boyd, S.: Semi-definite programming. SIAM Rev. **38**, 49–95 (1996)
13. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
14. Wolberg, W., Mangasarian, O.: Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc. Natl. Acad. Sci. **87**, 9193–9196 (1990)

Multi-Objective Optimization and Multi-Attribute Decision Making for a Novel Batch Scheduling Problem Based on Mould Capabilities

Jun Pei, Athanasios Migdalas, Wenjuan Fan, and Xinbao Liu

Abstract This chapter investigates a novel multi-objective model of a batch scheduling problem with constraint of the mould capability, and the objective is to minimize both the total completion time of the jobs and the total cost of the moulds. It is extremely difficult to obtain an optimal solution to this type of complex problems in a reasonable computational time. In view of this, this chapter presents a new multi-objective algorithm based on the features of Gravitational Search Algorithm to find Pareto optimal solutions for the given problem. In the proposed algorithm a novel Pareto frontier adjustment strategy is designed and proven to improve the convergence of solutions and increase convergence speed. We examined a set of test problems to validate the high efficiency of the proposed multi-objective gravitational search algorithm based on a variety of metrics. Finally, a multi-attribute decision making method is employed to determine the trade-off solutions derived from the Pareto optimal set and thus solve the problem optimally.

J. Pei (✉)
Department of Information Management and Information Systems, School of Management, Hefei University of Technology, Hefei, China
e-mail: feiyijun198612@163.com

Department of Industrial and Systems Engineering, Center for Applied Optimization, University of Florida, Gainesville, FL, USA

A. Migdalas
Industrial Logistics, ETS Institute, Luleå University of Technology, Luleå, Sweden

Division of Transportation, Construction Management and Regional Planning, Department of Civil Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece
e-mail: athmig@ltu.se

W. Fan • X. Liu
Department of Information Management and Information Systems, School of Management, Hefei University of Technology, Hefei, China

Key Laboratory of Process Optimization and Intelligent Decision-Making of Ministry of Education, Hefei, China

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_14

Keywords Batch scheduling • Mould capability • Gravitational search algorithm • Technique for order preference by similarity to ideal solution (TOPSIS)

1 Introduction

Batch scheduling problems widely exist in many industrial areas, such as aluminum productions, logistics transportation, semiconductor manufacturing, and production in the chemistry industry. For batch scheduling problems, multiple jobs in a batch are processed by batching machine at the same time, and the total size of all jobs in one batch cannot exceed the capacity of the machine. Optimizing the utilization of space is usually not considered in traditional scheduling problems, while for batch scheduling problems, ordering in time and spatial organization are considered. Therefore, compared to traditional scheduling problems, batch scheduling problems are more complicated.

The batch production can be divided into two types: serial batch and parallel batch. Serial batches are related to a group of jobs processed on a machine one after another, while parallel batches require that a number of jobs within a batch go through a machine and are processed simultaneously [35]. In this chapter, we consider a novel serial batch scheduling problem, for solving which we propose a new mathematical model and the objective is to minimize the total completion time of the jobs and the total cost of the moulds. Our proposed problem is on the basis of the real production of a Chinese aluminum production enterprise. The characteristics of this problem are included as follows:

1. The jobs processing needs not only the machines but also the coordination of the moulds.
2. The utilization of the moulds is limited by their capacity, and they cannot be used after their capacity is achieved.
3. The serial jobs processed in the same mould on one machine can form several batches. The batch processing time is the sum of the processing time of all jobs in the batch.

Different from traditional serial batch scheduling problems, the jobs have twofold resource constraints of both machines and moulds when they are processed. Also, the jobs are batched based on mould capacity, and the sum of the capacity consumed by a batch cannot exceed the capacity of the moulds which are used to process the batch. For this complex batch scheduling problem, we propose an effective approach that combines multi-objective gravitational search algorithm (MOGSA) and multi-attribute decision making (MADM) method to solve it.

This chapter is organized as follows: We start Sect. 2 with a literature review. The modeling of multi-objective batch scheduling problem based on mould capability is described in Sect. 3. Section 4 provides and proves three properties of the optimal solutions. We propose the MOGSA algorithm in Sect. 5. The MADM

method is provided in Sect. 6. The computational experiments are conducted and the computational results are discussed in Sect. 7. We conclude the chapter in Sect. 8.

2 Literature Review

Previous literature related to our considered scheduling problem in this chapter can be classified into two categories. They are (1) batch scheduling problems, and (2) scheduling problems with mould constraint.

2.1 Batch Scheduling Problems

The batch scheduling problems include two types: batch scheduling problems with single-objective and batch scheduling problems with multi-objective, which are as follows:

2.1.1 Batch Scheduling Problems with Single-Objective

Several recent studies have provided methods to obtain solutions for serial batch scheduling problem with single-objective. Coffman et al. [5] provided an $O(nlogn)$ algorithm to solve the problem $1|s-batch|\sum C_j$. Albers and Bruker [1] presented an $O(n2)$ algorithm to solve the problem $1|prec;p_j=p;s-batch|\sum w_jC_j$. Webster and Baker [32] proposed an algorithm to solve the problem $1|s-batch|L_{max}$ in $o(n2)$ time. Baptice [3] gave an $O(n^14)$ algorithm to solve the problem. Ng et al. [19] reduced the problem $1|prec;s-batch|L_{max}$ to the problem $1|s-batch|L_{max}$ and used a revised Webster–Baker algorithm to solve it in $O(n2)$ time. Xuan and Tang [35] considered the problem of serial batch scheduling jobs in an s-stage hybrid flow shop at the last stage. And the objective is to minimize a given criterion with respect to the completion time, and the authors proposed a batch decoupling-based Lagrangian relaxation algorithm for this problem.

In recent years, there have been extensive studies of parallel batch scheduling problems with single-objective, using heuristic algorithms and/or intelligent algorithms. Potts and Kovalyov [26] conducted an extensive review in the field of parallel batching literature, gave details of the basic algorithms, and referenced to other significant results. Nong et al. [20] investigated the problem of single machine parallel batching to minimize the makespan by considering family setups and release date, and they developed a polynomial approximation scheme for their problem that yielded to an algorithm with worst case ratio of 2.5. Mirsanei et al. [18] considered the parallel batch scheduling problem of a two-stage flow shop with two batch processing machines to minimize the makespan and developed two different simulated annealing (SA) algorithms based on two constructive heuristics. Due to

the importance of on-time delivery in semiconductor manufacturing, Mathirajan et al. [17] studied the parallel batch scheduling problem with the objective of minimizing total weight and proposed a simulated annealing (SA) algorithm to solve it. Su and Chen [30] studied a two-machine flow shop problem in which a parallel batch processor is followed by a discrete processor, and presented a heuristic algorithm and a branch-and-bound algorithm. Manjeshwar [16] studied the problem $F2|p-batch|C_{max}$ and proposed a genetic algorithm (GA) to solve it. Damodaran et al. [6] proposed a particle swarm optimization (PSO) algorithm to solve the problem. In recent years, we also published some papers on the serial batch scheduling problems, where multiple manufacturers and deteriorating jobs were considered [21–25]

2.1.2 Batch Scheduling with Multi-Objective

Multi-objective batch scheduling problem has less been considered than single-objective problem. Chinchuluun and Pardalos [4] did a survey of recent developments in multi-objective optimization, including optimality conditions, applications, global optimization techniques, the new concept of epsilon Pareto optimal solution, and heuristics. Fontes and Gaspar-Cunha [9] discussed some details of multi-objective evolutionary algorithms. They gave a summary of the main algorithms behind approaches and applications, including advantages and disadvantages, degree of applicability, and some applications. Loukil et al. [15] discussed a production scheduling problem in a flexible job-shop with batch production, and they proposed a multi-objective simulated annealing approach to tackle this problem with four different objectives of the makespan, the mean completion time, the maximal tardiness and the mean tardiness, respectively. Li et al. [27] modeled the dry strip operations in a real wafer fab as a PBPM scheduling problem and used an Ant Colony Optimization (ACO) algorithm to simultaneously minimize the total weighted tardiness (TWT) and makespan of the jobs. Zhang et al. [36] studied a hot-rolling batch scheduling problem. Their goal was to minimize changes in the characteristics of all neighbor steel strips and maximize the machine utilization. They formulated this problem as a combinatorial multi-objective optimization problem and developed a new heuristic approach by enhancing the framework of PSO for the problem. Kashan et al. [14] investigated the problem of scheduling jobs with non-identical sizes on a single batch processing machine and proposed two different multi-objective genetic algorithms based on different representation schemes for bi-criteria minimization of makespan and maximum tardiness. Haddad et al. [12] proposed a new mathematical model for a serial batch scheduling problem to minimize the maximum lateness and delivery cost, and the authors gave simulation annealing meta-heuristic to solve it. Azzi et al. [2] studied the HFS scheduling problem in a flexible multi-stage batch production system, and they offered a heuristic procedure to minimize the production makespan and increase the productive capacity utilization using a batch aggregation and splitting strategy.

2.2 Scheduling Problems with Mould Constraint

Gao et al. [10] addressed the problem of scheduling job groups on identical parallel machines with single mould constraint to implement the objective of minimizing total tardiness penalties, and they proposed a heuristic run-based sequencing scheduling algorithm. Xi et al. [33] established a multi-resources-constrained scheduling optimization model with machine and mould constraints. According to this model, a three-level optimization heuristic algorithm including parts level, resources level, and optimization level was designed, and an example was illustrated to verify its feasibility and effectiveness. Based on the multi-agent system and the immunity information processing mechanism, they proposed a multi-agent immune algorithm to solve the problem. Hong et al. [13] focused on minimizing the makespan of identical-machines scheduling problems with mould constraints, and they designed an adjustment operator to fill up the empty time slot due to the mould constraint. Ren et al. [29] established a dual-resource (i.e., the machines and the moulds resources)-constrained job-shop scheduling problem model and employed a heuristic active algorithm combined with priority rules to give the solution. Xu et al. [34] gave the model of the fuzzy flexible Job-shop scheduling problem with a variety of batches, considering various resource constraints, including moulds, machines, operators, the uncertainty factors of processing time, and due date in the practical diffuser shops.

2.3 Summary

Based on the above brief review, we find that there is a gap between theory development and practical applications. Despite a broad body of literature on the batch scheduling problems, only few approaches have been implemented for real industrial applications. Besides, there is limited research work on multi-objective batch scheduling problems and most of them only consider the machine capacity, which is not always in accordance with the actual case. Only a few papers consider mould constraints, while most of them just consider the constraint by the number of moulds, i.e., at any given time one mould can only process one job, and the total number of moulds used by jobs cannot exceed the inventory number of the moulds. However, in the real production, the mould usage may not be only reflexed by their number, but also by their capacity, which are found in the literature. To the best of our knowledge, our work is the first effort on studying the scheduling problems which consider the mould capacity constraint. This research is also an attempt to bridge the gap between theory development and practice applications by proposing a comprehensive solution to this practical case that can integrate all characteristics.

3 Problem Description and Modelling

3.1 Notation

Parameters:

i	index of the machines, the ith machine is denoted as M_i.
j	index of the jobs, the jth job is denoted as J_j.
k	index of the batches, the kth batch is denoted as b_k.
h	index of the moulds, the hth *mould* is denoted as H_h.
n	total number of jobs.
m	total number of machines.
s	total number of moulds.
p	processing time of each job.
e	cost of each mould.
n_k	number of jobs in the batch b_k.
w_j	weight of the job J_j.
B_i	set of all batches processed on the machine M_i.
K_i	number of batches processed on the machine M_i.
d_k	number of moulds which are used to process the batch b_k.
p_{ik}	processing time of the batch b_k processed on the machine M_i.
K	total number of batches, i.e., $K = K_1 + K_2 + \ldots_+ K_m$.

Decision variables:

x_{ij}	1, if the machine M_i processes the job J_j; 0, otherwise.
y_{ih}	1, if the machine M_i uses the mould H_h; 0, otherwise.
z_{jk}	1, if the job J_j belongs to the batch b_k; 0, otherwise.
g_{jh}	1, if the mould Hh processes the job J_j; 0, otherwise.
T_i	the completion time of the jobs processed on the machine M_i.
C_j	the completion time of the job J_j.

3.2 Problem Description

Definition 1. The capacity of the mould means the maximum total weight of jobs that a single mould can process continuously.

In this chapter, the capacity of each mould is set fixed, denoted as a. Suppose there are n jobs which can be processed on m parallel machines, and their processing time lasts in a same period, denoted as p. During the job processing, the moulds are required to cooperate with the machines. The weight of each job may be different.

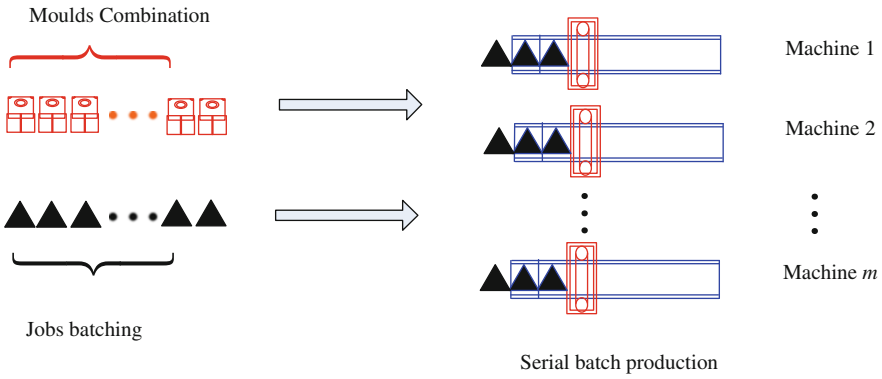


Fig. 1 The layout of the batch scheduling problem based on mould capabilities

The jobs are divided into several batches based on moulds before being processed. A single mould can process no more than one batch of job at once. The moulds can be combined to process a batch of the jobs. Thus, the total mould capacity in a possible batch may change when multiple moulds are combined. The jobs in each batch can be randomly processed on no more than one machine. The processing time of the batch b_k is represented by p^k, and $p^k = \sum_{J_j \in b_k} p_j$. The start and completion times of processing the batch b_k are denoted by s^k and c^k, and we have $c^k = s^k + p^k$. The completion time of each job in a batch is equal to its batch completion time, i.e., $c_j = c^k (J_j \in b_k)$. If a mould is unloaded from a machine, it will not be used any more. The layout of the scheduling problem is shown in Fig. 1. The model assumptions are summarized as follows:

1. All the resources (moulds and machines) are all available at time zero in the usage time.
2. The mould resources can meet the task requirements for production scheduling.
3. The mould capacity consumed by a single job is no larger than the corresponding mould capacity.
4. Preemption is prohibited, i.e., once the processing of a batch has begun, it cannot be stopped.

In order to differentiate our problem from previous batch scheduling problem, we can denote the problem as $P|\sum_{J_j \in b_k} w_j \leq n \cdot a, p_j = p, s - batch|\sum_{j=1}^{n} C_j, \sum_{k=1}^{K} ed_k$ by adopting the three-field notion $\psi_1|\psi_2|\psi_3$ of Graham et al. [11], where P denotes the machines of the same type, $\sum_{J_j \in b_k} w_j \leq n \cdot a$ denotes that the mould capacity consumed by the jobs in the same batch is no larger than the total capacity of the moulds which are used to process the batch, $p_j = p$ denotes that the processing time of the jobs are the same, $s - batch$ denotes that it is a serial batch scheduling problem,

$\sum_{j=1}^{n} C_j$ denotes the total completion time of jobs, and $\sum_{k=1}^{K} ed_k$ denotes the total cost of moulds. For simplicity, we use to represent our proposed problem. There are similarities between the batch scheduling problems based on mould capacity and traditional ones. However, there is still an essential difference in the way of batching. In the traditional batch scheduling problems, if the maximum capacity space of the machine is assumed to be a constant B, then the total size of the jobs in each batch is no larger than B, and the machine capacity constraint is only considered during the jobs processing. In our problem, if the number and capacity of the moulds are assumed to be two constants M and a, respectively, then any one mould from M moulds can be combined together to process a batch. We have to consider not only the mould capacity constraint in job batches processing but also the combination of moulds, i.e., the mould capacity for possible batches may be $a, 2a, \ldots, (M-1)a$, and Ma. Therefore, the number of jobs in a batch depends on the number and capacity of used moulds.

3.3 Mixed Integer Programming Model

$$\textit{Minimize} \qquad f_1 = \sum_{j=1}^{n} C_j \tag{1}$$

$$\textit{Minimize} \qquad f_2 = \sum_{k=1}^{K} ed_k \tag{2}$$

Subject to

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j = 1,2,\ldots,n \tag{3}$$

$$\sum_{k=1}^{K} z_{jk} = 1 \qquad j = 1,2,\ldots,n \tag{4}$$

$$\sum_{i=1}^{m}\sum_{h=1}^{s} y_{ih} \le s \tag{5}$$

$$\sum_{j=1}^{n} w_j \le s \cdot a \tag{6}$$

$$\sum_{j=1}^{n} w_j q_{jh} \le a \qquad h = 1,2,\ldots,s \tag{7}$$

$$x_{ij}, y_{ih} z_{jk}, q_{jh} \in \{0,1\} \quad i = 1,2,\ldots,m; j = 1,2,\ldots,n; \quad k = 1,2,\ldots,K; h = 1,2,\ldots,s \tag{8}$$

Objective function (1) minimizes the total completion time of jobs. Objective function (2) minimizes the total cost of the used moulds. Constraint (3) specifies that one machine cannot process more than one job at one time. Constraint (4) guarantees that any one job should belong to only a batch. Constraint (5) ensures that the total number of the moulds used by all machines is less than the inventory number of the moulds. Constraint (6) indicates that the total weight of the jobs is no more than the total capacity of the moulds in inventory. Constraint (7) ensures the total weight of the jobs processed by one mould is no larger than its mould capacity. Constraint (8) defines the range of the variables.

4 Properties of Optimal Solutions

An optimal solution is denoted as Λ^*. The optimal function values of the total job completion time and the total cost of the used moulds in Λ^* are represented as U_1^* and U_2^*, respectively. The number of batches processed on the machine M_i in Λ^* is represented as d_i. The residual capability of the moulds which are used to process the kth batch b_k is denoted as r_k. The minimum job weight of all jobs in the batch b_k is denoted as w^k.

Property 15.1. In Λ^*, the optimal solution will remain unchanged when any two batches processed on the machine $M_i (i = 1, 2, \ldots, m)$ are swapped.

Proof. Suppose there are two batches b_k and $b_{k'}$ processed on the machine in the optimal solution. The total processing time of the jobs on all machines except the machine m_i is denoted as c, the total processing time of the jobs processed on the machine M_i except the batches b_k and $b_{k'}$ is denoted as c', and the number of the jobs processed between the two batches on the machine M_i is denoted as n_x. When $s^k > s^{k'}$, the optimal value of f_i is $U^* = c + c' + n_k(s^k + n_k p) + n_{k'}(s^k + n_k p + n_{k'} p + n_x p)$. A new solution Λ' is generated after swapping the batches b_k and $b_{k'}$, and the updated function value of f_1 is $U' = c + c' + n_{k'}(s^k + n_{k'} p) + n_k(s^k + n_k p + n_{k'} p + n_x p)$. It is obtained that $U' - U^* = 0$. Similarly, when $s^k < s^{k'}$, it can be also inferred that $U' - U^* = 0$. Thus, the new solution Λ^* is still the optimal solution, and the proof is completed.

Property 15.2. In Λ^*, if $\exists b_k \subseteq B_i, b_{k'} \subseteq B_i$, and $w^{k'} < r_k$, then $n_{k'} - n_k \leq 1$.

Proof. The total processing time of the jobs on all machines except the machine M_i is denoted as c, the total processing time of the jobs on the machine M_i except the batches b_k and $b_{k'}$ is denoted as c', and the number of jobs processed between the two batches on M_i is n_x. When $s^k > s^{k'}$, there is $U_1^* = c + c' + n_x(s^k + n_k p) + n_{k'}(s^k + n_k p + n_x p + n_{k'} p)$. Because $w^{k'} < r_k$, J_j can be transferred from the batch $b_{k'}$ to the batch b_k. The processing batches on the machine M_i become $(\ldots, b_k \cup \{J_j\}, \ldots, b_{k'} \backslash \{J_j\}, \ldots)$ after interchanging, and a new solution Λ' is created. Since the mould cost remains unchanged, we get that $U_2' = U_2^*$.

Furthermore, $U_1' = c + c' + (n_k + 1)[s^k + (n_k + 1)p)] + n_x p + n_{k'} - 1)[s^k + (n_k + 1)p + n_x p + (n_{k'} - 1)p]$, and it can be inferred that $U_1^* - U_1' = (n_{k'} - n_k - 1)p$. Because Λ^* is an optimal solution, it is obtained that $U_1^* - U_1' \leq 0$, which means $(n_{k'} - n_k - 1)p \leq 0$. Because $p > 0$, we get that $n_{k'} - n_k \geq 1$. Similarly, when $s^k \leq s^{k'}$, the conclusion is also satisfied. Therefore, the proof is completed.

5 The Proposed MOGSA

5.1 Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is an optimal search algorithm based on simulation to gravity in physics proposed by E. Rashedi et al. [28]. In GSA, the mass of individual depends on the quality of solution, i.e., the better the solution, the larger the individual mass. It is to search optimal solution by seeking the individual with maximum mass. The movement of individuals follows Newton's second law, so each individual will move towards the one with maximum mass because of gravity, and their location will also change. In the end, all the individuals will gather around the individual with maximum mass, so that the optimal solution will be found. According to E. Rashedi et al. [28], the mass of individual X_p is defined by

$$m_p(l) = \frac{fit_p(l) - worst(l)}{best(l) - worst(l)} \tag{9}$$

$$M_p(l) = \frac{m_p(l)}{\sum_{p=1}^{N} m_p(l)} \tag{10}$$

where $p = 1, 2, \ldots, N4$, and N represents the number of individuals, $m_p(l)$ and $M_p(l)$ indicate the function value and mass of individual X_p in the lth iteration, respectively, and the optimal and worst function values of all individuals in the lth iteration are denoted as $best(l)$ and $worst(l)$, which are defined as Eqs. (11) and (12) (E. Rashedi et al. [28]):

$$best(l) = \min_{p \in \{1,2,\ldots,N\}} fit_p(l) \tag{11}$$

$$worst(l) = \min_{p \in \{1,2,\ldots,N\}} worst_p(l) \tag{12}$$

In the dth dimension, the gravity formula between individual X_p and X_q is given as follows (E. Rashedi et al. [28]):

$$F_{pq}^{d}(l) = G(l)\frac{M_p(l)M_q(l)}{R_{pq}(l)+\varepsilon}(x_{qd}^{l} - x_{pd}^{l}) = G_{0e}^{-\alpha\frac{l}{T}}\frac{M_p(l)M_q(l)}{R_{pq}(l)+\varepsilon}(x_{qv}^{l} - x_{pv}^{l}) \tag{13}$$

where $R_{pq}(l)$ represents the Euclidean distance between individuals X_p and X_q in the lth iteration; ε is a tiny constant; $G(l)$ denotes the gravitational constant in the lth iteration; both G_0 and α are constants; T represents the maximum number of iterations; x_{qv}^{l} and x_{pv}^{l} represent the location of individuals X_p and X_q on the dth dimension in the lth iteration, respectively.

The accelerated speed of individual X_p on the dth dimension in the lth iteration can be denoted as Eq. (14) (E. Rashedi et al. [28]):

$$a_{pd}^{l} = \frac{F_p^d(l)}{M_p(k)} = \frac{\sum_{q=1,q\neq p} randF_{pq}^{d}(l)}{M_p(k)} \tag{14}$$

where $F_p^d(l)$ represents the resulting force on individual X_p on the dth dimension in the lth iteration; *rand* is a random number in $[0,1]$, which follows uniform distribution. The update formula of the speed and location of individual X_p are as Eqs. (15) and (16) (E. Rashedi et al. [28]):

$$v_{pd}^{l+1} = randv_{pd}^{l} + a_{pd}^{l} \tag{15}$$

$$x_{pd}^{l+1} = x_{pd}^{l} + v_{pd}^{l+1} \tag{16}$$

Based on GSA, we propose MOGSA to solve the multi-objective optimization problem.

5.2 *Key Procedure of MOGSA*

5.2.1 Encoding

Previous encoding schemes mainly just mark the individuals of the jobs and hardly reflect the grouping of the jobs. Aiming to this problem, we propose an encoding scheme of two n-dimension vectors, which are denoted as A and B vectors, respectively. The element values in A vector are real numbers, and each element represents a job number of which the sequence indicates the processing order of the job. The element values in B vector are equal to 0 or 1, and the corresponding jobs between two elements of value 1 can form a batch. The first and last element values

Table 1 Encoding example

T	1	2	3	4	5	6	7	8	9	10
A vector	7	8	5	1	6	4	9	2	10	3
B vector	1	0	0	1	0	1	0	0	0	0

Table 2 Individual encoding modification

T	1	2	3	4	5	6	7	8	9	10
A vector before iteration	7	8	5	1	6	4	9	2	10	3
B vector before iteration	1	0	0	1	0	1	0	0	0	0
A vector after iteration	9.2	1.1	8.7	4.2	4.5	0.9	6.7	5.5	4.9	3.6
B vector after iteration	1	0.3	0.2	0.6	0.7	0.2	0.8	0.4	0.3	0
A vector before modification	10	2	9	4	5	1	8	7	6	3
B vector before modification	1	0	0	1	1	0	1	0	0	0

in B vector are set as 1 and 0, respectively. Table 1 provides a simple encoding example. As shown in Table 1, the first job batches are $\{7,8,5\}$, the second job batches are $\{1,6\}$, and the third job batches are $\{4,9,2,10,3\}$.

5.2.2 Decoding

When GSA is applied in the scheduling problems, some illegal solutions may be generated. Thus, these illegal solutions need to be modified. For A vector, we apply decreasing decoding method to modify it, i.e., sorting the values in A vector after iteration in descending order and then replacing them with the job numbers. With respect to B vector, we use the method of replacing similar values to modify it. If the value of B vector after iteration is smaller than 0.5, it will be replaced by 0; otherwise it will be replaced by 1. In Table 2, the detailed adjustments of one example are shown. For A vector, jobs are sorted in descending order. The first time we replace the maximum value 9.2 with job 10, the second time we replace the second maximum value 8.7 with job 9, and so on. For B vector, the first element remains unchanged, and second element becomes 0, and so on.

In the process of decoding, the job batching is determined by B vector. The processing sequence of the batches is based on their generation order. The rule of selecting machine for the batches is to select the machine which can start earliest. The objective function values are calculated based on Eqs. (1) and (2). The mould cost in batch b_k is calculated by Eq. (17):

$$f(k) = e \ \mathit{floor}\left(\sum_{J_j \in b_k} w_j / a\right) \tag{17}$$

where $\mathit{floor}()$ is a floor function.

5.2.3 Selector Operator

After the two objective function values of each individual are determined, we apply fast non-dominated sort algorithm [8] to divide rank of Pareto solution set of each individual. Then, we apply the method proposed by Davis [7] to compute the individual congestion degree, and each individual X_p in population has two attributions after fast non-dominated sort and congestion degree calculation: domination level attribution p_{rank} and congestion distance attribution p_{dis}. The rule of the individual comparison operator is as follows: if $p_{rank} > q_{rank}$ or $p_{rank} = q_{rank}$ and $p_{dis} > q_{dis}$, then keep the individual X_p and eliminate the individual X_q, otherwise keep the individual X_q and eliminate the individual X_p.

We apply the best individual preservation strategy to increase the search speed of the algorithm add a best individual X_{best} to keep the best individual in each iteration. Let F_t denote the non-dominated individual set of the tth level. The selection of X_{best} in the lth iteration is as follows:

1. If the number of individuals in F_1 is 1, then X_{best} is set as the individual included in F_1.
2. If the number of individuals in F_1 is larger than 1, then each individual's standard value of congestion degree is calculated. There is an individual $X_q \in F_1$, and the calculation of the congestion degree of the individual X_q's standard value is as Eq. (18):

$$\lambda_q = \frac{q_{dis} - \min\limits_{X_p \in F_1} p_{dis}}{\max\limits_{X_p \in F_1} p_{dis} - \min\limits_{X_p \in F_1} p_{dis}} \tag{18}$$

Select an individual X_q from F_1 and generate a random number $rand \in [0, 1]$. If $rand \leq \lambda q$, then set X_{best} as X_q, otherwise repeat the process. If the selected individual's congestion degree is the largest in F_1, then the standard value of congestion degree is 1 by Eq. (18), and the individual must be set as X_{best}. At the same time, other individuals may be selected as X_{best}. This keeps the diversity of X_{best}. X_{best} and other N individuals involve the $(l+1)$th iteration.

5.2.4 Mutation Operator

In order to improve global search ability of the proposed algorithm, we introduce mutation operator here, i.e., to proceed mutation on the B vector of all individuals. The probability of mutation on individuals in F_t is calculated as Eq. (19):

$$\mu_t = 0.02 + 0.08 \times \exp\left(\frac{t}{t_{\max}} - 1\right) \tag{19}$$

where $t_{\max}$ is the maximum number of levels in Pareto set. In respect to each individual X_p in F_t, generate a random number $rand \in [0,1]$. If $rand \leq \mu_t$, then proceed mutation, and select randomly a dimension of value in B vectors of individual X_p, if the value is 1, then shift to 0, otherwise 0 to 1. The larger the number of individual's level, the larger the probability of mutation. Mutation operation avails to find better individual for dominated solutions.

5.2.5 Pareto Frontier Adjustment Strategy

Based on the Property 2 of optimal solutions, we should adjust the batches of each solution in Pareto frontier. In respect to some solution, we select two batches b_k and $b_{k'}$ processed on a machine randomly. If $w^{k'} < r_k$ and $n_{k'} - n_k < 1$, then transfer the lightest job from the batch $b_{k'}$ to the batch b_k until $w^{k'}$ or $n_{k'} \geq 1$.

5.3 Algorithm Processes of MOGSA

A detailed flowchart of the proposed MOGSA is shown in Fig. 2.

6 Multi-Attribute Decision Making

Since production staff in enterprises usually follow and execute only one decision plan, we need to select an ideal solution from Pareto frontier solution set. This chapter applies TOPSIS method (Technique for Order Preference by Similarity to an Ideal Solution) to do MADM [31]. The detailed procedures are as follows:

Step 1 Build decision making matrix based on Pareto optimal solution set [31].

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{m1} & x_{m2} \end{bmatrix} \tag{20}$$

where x_{ij} denotes the attribution value of the jth objective evaluation attribute corresponding to the ith solution in Pareto optimal solution set, and m represents the number of solutions in Pareto optimal solution set.

Step 2 Standardize the decision making matrix.

Because there is incommensurability between each objective, we need to

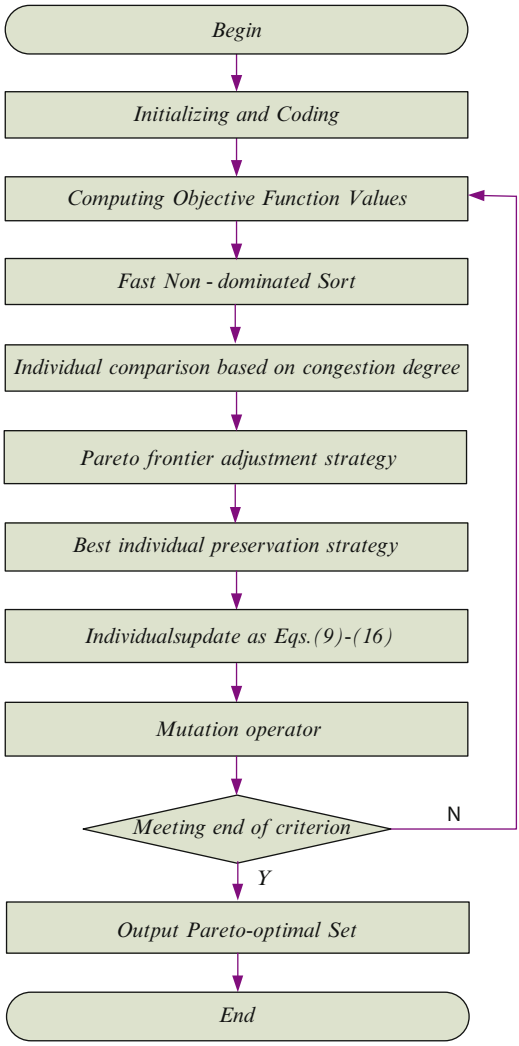


Fig. 2 The flowchart of MOGSA

standardize all the objective value before evaluation. Since the two objective functions in this batch scheduling problem we discuss are both cost type, the standardization formula is as Eq. (21) [31]:

$$y_{ij} = \begin{cases} \dfrac{\max\limits_j x_{ij} - x_{ij}}{\max\limits_j x_{ij} - \min\limits_j x_{ij}} & \max\limits_j x_{ij} \neq \min\limits_j x_{ij} \\ 1 & \max\limits_j x_{ij} = \min\limits_j x_{ij} \end{cases} \tag{21}$$

where $i = 1,2,\ldots,m$, y_{ij} denotes the attribution value of the jth objective evaluation attribute corresponding to the ith solution, and $y_{ij} \in [0,1]$.

Step 3 Determine the weighted standardized decision making matrix.

$$Z = (z_{ij})_{m\times 2}$$

where $z_{ij} = \omega_j y_{ij}, i = 1,2,\ldots,m$, and $j = 1,2$.

We apply entropy weighting method to determine the weight w_j of each attribute, and the basic processes are as follows [31]:

1. Compute the entropy value H_j of the jth objective evaluation attribute, and the calculation is as Eq. (22):

$$H_j = -\frac{1}{\ln m}\sum_{i=1}^{m} f_{ij}, j = 1,2 \tag{22}$$

where $f_{ij} = \dfrac{y_{ij}}{\sum_{i=1}^{m} y_{ij}} i = 1,2,\ldots,m$, and $j = 1,2$

2. Compute weight according to the entropy value of jth objective evaluation attribute:

$$\omega_j = \frac{1 - H_j}{2 - \sum_{j=1}^{2} H_i} \tag{23}$$

Step 4 Calculate ideal solution and negative ideal solution [31]:

$$Z^+ = \{z_1^+, z_2^+\} = \{\max_i z_{ij} | j = 1,2\} \tag{24}$$

$$Z^- = \{z_1^-, z_2^-\} = \{\min_i z_{ij} | j = 1,2\} \tag{25}$$

Step 5 Compute the Euclidean distance between each objective function and ideal solution and negative ideal solution as Eqs. (26) and (27) [31], and they are denoted as d_i^+ and d_i^-, respectively.

$$d_i^+ = \|z_i - z^+\| = \sqrt{\sum_{j=1}^{2}(z_i^+ - z_{ij})^2}, i = 1,2,\ldots,m \tag{26}$$

$$d_i^- = \|z_i - z^-\| = \sqrt{\sum_{j=1}^{2}(z_{ij} - z_j^-)^2}, i = 1,2,\ldots,m \tag{27}$$

Step 6 Compute the relative nearness degree of each objective function and ideal solution as Eq. (28) [31]:

$$c_i = d_i^- / (d_i^+ + d_i^-), i = 1, 2, \ldots, m \tag{28}$$

Step 7 Sort the relative nearness degree c_i by descending order, and the former one is better than the latter one. We choose the solution which has the largest relative nearness degree c_i as the final scheduling plan.

7 Experimental Results

7.1 Test Problems

The experiment contains 9 test problems of different sizes generated, which is presented in Table 3. The other parameters are randomly generated based on the real aluminum production as follows: Weight of the jobs $w_j (j = 1, 2, \ldots, m)$ generated from the continuous uniform distribution $U = [3, 23]$. Processing time of each job p generated from the continuous uniform distribution $U = [0.05, 0.2]$. Cost of each mould generated e from the discrete uniform distribution $U = [5, 15]$. Capacity of each mould a generated from the discrete uniform distribution $U = [28, 32]$. The proposed MOGSA is applied to these problems and we use a number of comparison metrics to assess its performance.

7.2 Results of MADM and the Schedule's Gantt Chart Corresponding to Optimal Solutions

Tables 4, 5, and 6 display the results of non-dominated solutions obtained by MOGSA and the values of similarity degree obtained by TOPSIS method of 9 problems. Based on the values of similarity degree in the 9 problems, the solutions

Table 3 Size characteristics of 9 problem sets

Problem	Jobs (n)	Machines (m)
1	30	3
2	30	4
3	30	5
4	60	3
5	60	4
6	60	5
7	90	3
8	90	4
9	90	5

Table 4 Results of non-dominated solutions obtained by MOGSA

Problem 1			Problem 2			Problem 3		
f_1	f_2	c_i	f_1	f_2	c_i	f_1	f_2	c_i
17.1	250	0.5	13.4	250	0.52	11.1	250	0.52
17.3	240	0.55	13.6	240	0.55	11.3	240	0.55
17.4	230	0.58	13.7	230	0.58	11.4	230	0.58
17.6	210	0.67	13.8	220	0.62	11.6	220	0.62
18	200	0.71	13.9	210	0.66	11.7	210	0.66
18.4	190	0.74	14.2	200	0.7	12	200	0.7
18.7	180	0.78	14.7	190	0.73	12.4	190	0.74
19.9	170	0.7	15	180	0.76	12.8	180	0.76
23.6	160	0.48	16.2	170	0.69	13.9	170	0.71
			19.8	160	0.48	18	160	0.48

Table 5 Results of non-dominated solutions obtained by MOGSA (continued)

Problem 4			Problem 5			Problem 6		
f_1	f_2	c_i	f_1	f_2	c_i	f_1	f_2	c_i
65.1	470	0.52	50.1	470	0.51	41.2	470	0.52
65.3	460	0.54	50.4	460	0.53	41.4	460	0.54
65.5	450	0.56	50.5	450	0.55	41.6	450	0.56
65.6	440	0.58	50.6	440	0.57	41.7	440	0.58
65.8	430	0.61	50.9	430	0.6	41.9	430	0.61
66.1	420	0.63	51.1	420	0.63	42.1	420	0.64
66.5	410	0.66	51.1	420	0.63	42.5	410	0.67
66.9	400	0.69	51.5	410	0.66	42.9	400	0.7
67.6	390	0.72	51.8	400	0.69	43.6	390	0.72
68.5	380	0.73	52.6	390	0.71	44.6	380	0.73
68.8	370	0.75	53.7	380	0.72	45	370	0.75
69.4	360	0.76	53.9	370	0.75	45.5	360	0.76
71.8	350	0.67	54.4	360	0.76	48	350	0.68
79.1	340	0.48	57	350	0.68	55.5	340	0.48
			64.6	340	0.49			

with maximum c_i of each problem are taken as the satisfactory solutions. The schedule's Gantt chart corresponding to optimal trade-off solutions of 9 problems are shown from Figs. 3, 4, 5, 6, 7, 8, 9, and 10.

8 Conclusions

This chapter presents a MOGSA to solve a novel batch scheduling problem with mould capacity constraint to minimize the total completion time of jobs and the total cost of moulds. We analyze this problem through proving three properties of optimal solutions, based on which the solutions in Pareto frontier are adjusted

Table 6 Results of non-dominated solutions obtained by MOGSA (continued)

Problem 7			Problem 8			Problem 9		
f_1	f_2	c_i	f_1	f_2	c_i	f_1	f_2	c_i
143.1	750	0.5	109.4	750	0.5	89.1	750	0.52
143.2	720	0.54	109.5	720	0.54	89.3	720	0.55
143.5	710	0.54	109.8	710	0.55	89.4	710	0.57
143.7	690	0.57	110	680	0.6	89.7	690	0.59
143.8	680	0.59	110.3	670	0.61	89.8	680	0.61
144.1	650	0.65	110.5	640	0.68	90.1	670	0.62
144.2	640	0.67	111.5	620	0.69	90.2	650	0.66
145.2	630	0.65	111.7	610	0.71	90.3	640	0.68
145.3	620	0.67	112.2	600	0.7	91.2	630	0.67
145.5	610	0.68	113.2	590	0.67	91.3	620	0.69
145.9	600	0.68	114.1	580	0.63	91.6	610	0.7
146.9	590	0.64	115.1	570	0.59	92.1	600	0.69
147.9	580	0.59	116.4	560	0.54	92.9	590	0.66
148.7	570	0.56	116.8	550	0.54	93.9	580	0.62
150	560	0.51	118.3	540	0.5	94.9	570	0.57
150.4	550	0.51				96.4	560	0.52
151.1	540	0.5				96.5	550	0.53
						98.2	540	0.48

in algorithm iteration. Various problems are experimented based on the proposed MOGSA. By the MADM method, decision-makers can obtain satisfactory solutions of each problem from the Pareto optimal set.

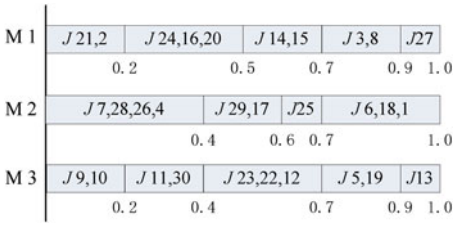


Fig. 3 Optimal trade-off solution 1 of problem 2

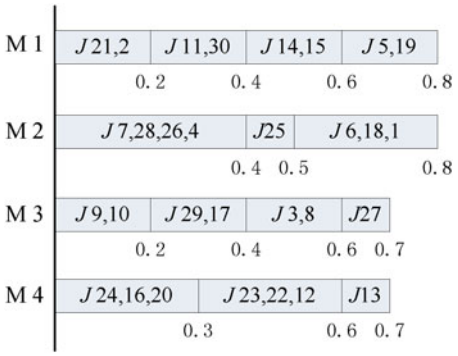


Fig. 4 Optimal trade-off solution 1 of problem 3

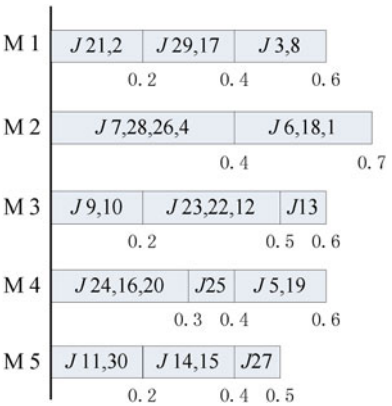


Fig. 5 Optimal trade-off solution 1 of problem 4

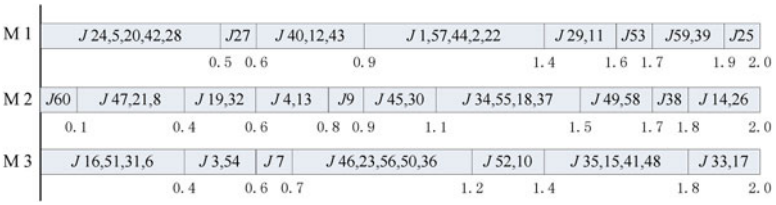


Fig. 6 Optimal trade-off solution 1 of problem 5

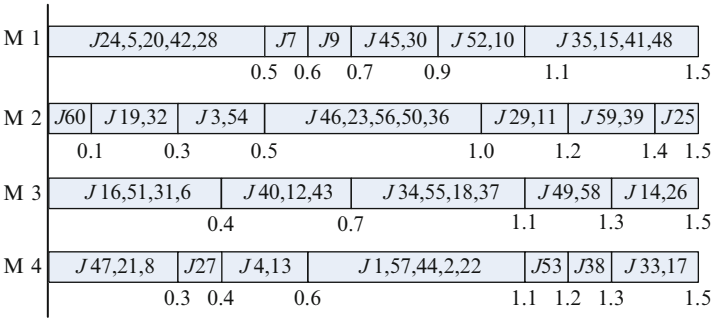


Fig. 7 Optimal trade-off solution 1 of problem 6

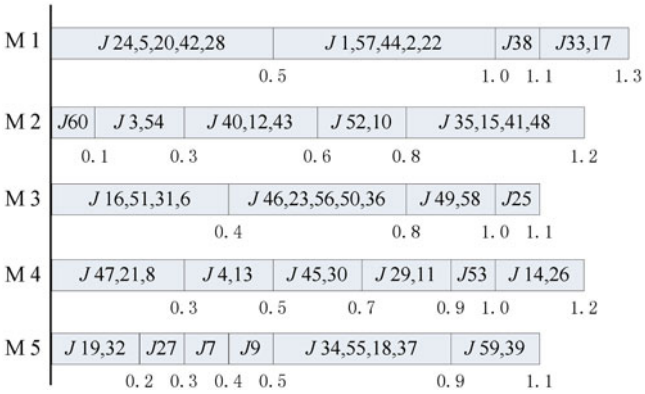


Fig. 8 Optimal trade-off solution 1 of problem 7

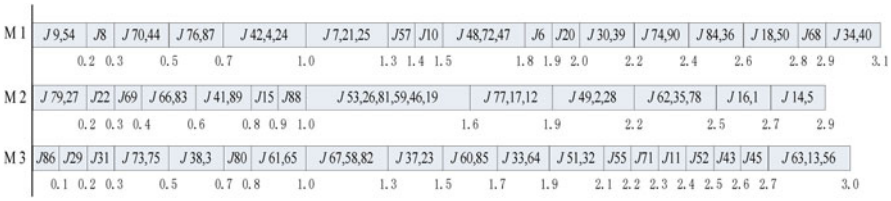


Fig. 9 Optimal trade-off solution 1 of problem 8

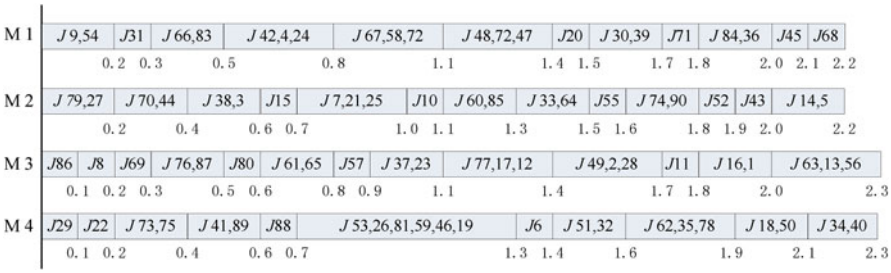


Fig. 10 Optimal trade-off solution 1 of problem 9

References

1. Albers, S., Brucker, P.: The complexity of one-machine batching problems. Discret. Appl. Math. **47**(2), 87–107 (1993)
2. Azzi, A., Faccio, M., Persona, A., Sgarbossa, F.: Lot splitting scheduling procedure for makespan reduction and machine capacity increase in a hybrid flow shop with batch production. Int. J. Adv. Manuf. Technol. **59**(5–8), 775–786 (2012)
3. Baptiste, P.: Batching identical jobs. Math. Meth. Oper. Res. **52**, 355–367 (2000)
4. Chinchuluun, A., Pardalos, P.M.: A survey of recent developments in multiobjective optimization. Ann. Oper. Res. **54**(1), 29–50 (2007)
5. Coffman, E.G., Yannakakis, M., Magazine, M.J., Santos, C.: Batch sizing and job sequencing on a single machine. Ann. Oper. Res. **2**(1), 135–147 (1990)
6. Damodaran, P., Diyadawagamage, D.A., Velez-Gallego, M.C., Ghrayeb, O.: A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines. Int. J. Adv. Manuf. Technol. **58**(9–12), 1131–1140 (2012)
7. Davis, L.: Hand Book of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
8. Deb, K., Pratap, A., Agrawal, S.: A fast and elitist multi objective genetic algorithm: Nsga-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
9. Fontes, D.B.M.M., Gaspar-Cunha, A.: On multi-objective evolutionary algorithms. In: Zopounidis, C., Pardalos, P.M. (eds.) Handbook of Multicriteria Analysis. Applied Optimization, vol. 103, chap. 10, pp. 287–310. Springer, Berlin (2010)
10. Gao, L., Wang, C., Wang, D., Yin, Z., Wang, S.: Heuristic to schedule grouped jobs on parallel machines with mould constraint. Control Decis. **14**(5), 392–397 (1999)
11. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic machine scheduling: a survey. Ann. Discrete Math. **5**, 287–326 (1979)
12. Haddad, H., Ghanbari, P., Moghaddam, A.Z.: A new mathematical model for single machine batch scheduling problem for minimizing maximum lateness with deteriorating jobs. J. Ind. Eng. Comput. **3**(2), 253–264 (2012)
13. Hong, T., Sun, P., Jou, S.: Evolutionary computation for minimizing makespan on identical machines with mold constrains. WSEAS Trans. Syst. Control **4**(7), 339–348 (2009)
14. Kashan, A.H., Karimi, B., Jolai, F.: An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes. Eng. Appl. Artif. Intell. **23**(6), 911–922 (2010)
15. Loukil, T., Teghem, J., Fortemps, F.: A multi-objective production scheduling case study solved by simulated annealing. Eur. J. Oper. Res. **179**(3), 709–722 (2007)

16. Manjeshwar, P.K., Damodaran, P., Srihari, K.: Genetic algorithms for minimizing makespan in a flow shop with two capacitated batch processing machines. Int. J. Adv. Manuf. Technol. **55**(9–12), 1171–1182 (2011)
17. Mathirajan, M., Bhargav, V., Ramachandran, V.: Minimizing total weighted tardiness on a batch-processing machine with non-agreeable release times and due dates. Int. J. Adv. Manuf. Technol. **48**(9–12), 1133–1148 (2010)
18. Mirsanei, H.S., Karimi, B., Jolai, F.: Flow shop scheduling with two batch processing machines and nonidentical job sizes. Int. J. Adv. Manuf. Technol. **45**(5–6), 553–572 (2009)
19. Ng, C.T., Cheng, T.C.E., Yuan, J.J.: A note on the single machine serial batching scheduling problem to minimize maximum lateness with precedence constraints. Oper. Res. Lett. **30**(1), 66–68 (2002)
20. Nong, Q., Ng, C.T., Cheng, T.C.E.: The bounded single-machine parallel-batching scheduling problem with family jobs and release dates to minimize makespan. Oper. Res. Lett. **36**(1), 61–66 (2008)
21. Pei, J., Fan, W., Pardalos, P. M., Liu, X., Goldengorin, B., Yang, S.: Preemptive scheduling in a two-stage supply chain to minimize the makespan. Optim. Methods Softw. (2014). (Online) doi:10.1080/10556788.2014.969262
22. Pei, J., Liu, X., Pardalos, P.M., Fan, W., Wang, L., Yang, S.: Solving a supply chain scheduling problem with non-identical job sizes and release times by applying a novel effective heuristic algorithm. Int. J. Syst. Sci. (2014). (Online) doi:10.1080/00207721.2014.902553
23. Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S., Wang, L.: Application of an effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain. Int. J. Adv. Manuf. Technol. **70**(1–4), 335–348 (2014)
24. Pei, J., Pardalos, P.M., Liu, X., Fan, W., Yang, S.: Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan. Eur. J. Oper. Res. (2014). (Online) doi:10.1016/j.ejor.2014.11.034
25. Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S.: Single machine serial-batching scheduling with independent setup time and deteriorating job processing times. Optim. Lett. **9**(1), 91–104 (2015)
26. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: a review. Eur. J. Oper. Res. **120**(2), 228–249 (2000)
27. Qiao, L.L.F., Wu, Q.D.: Aco-based multi-objective scheduling of parallel batch processing machines with advanced process control constraints. Int. J. Adv. Manuf. Technol. **44**(9–10), 985–994 (2009)
28. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)
29. Ren, H., Jiang, L., Xi, X., Li, M.: Heuristic optimization for dual-resource constrained job shop scheduling. In: 2009 International Asia Conference on Informatics in Control, Automation and Robotics, pp. 485–488 (2009)
30. Su, L.H., Chen, J.C.: Sequencing two-stage flowshop with non-identical job sizes. Int. J. Adv. Manuf. Technol. **47**(1–4), 259–268 (2010)
31. Wang, T.C., Lee, H.D.: Developing a fuzzy topsis approach based on subjective weights and objective weights. Expert Syst. Appl. **36**(5), 8980–8985 (2009)
32. Webster, S.T., Baker, K.R.: Scheduling groups of jobs on a single machine. Oper. Res. **43**(4), 692–703 (1995)
33. Xi, X., Jiang, L., Zhang, Q.: Optimization for multi-resources-constrained job shop scheduling based on three-level heuristic algorithm. In: 2009, International Asia Conference on Informatics in Control, Automation and Robotics, pp. 296–300 (2009)
34. Xu, X., Ying, S., Wang, W.: Fuzzy flexible job-shop scheduling method based on multi-agent immune algorithm. Control Decis. **25**(2), 171–178 (2010)
35. Xuan, H., Tang, L.: Scheduling a hybrid flowshop with batch production at the last stage. Comput. Oper. Res. **34**(7), 2718–2733 (2007)
36. Zhang, T., Chaovalitwongse, W.A., Zhang, Y.J., Pardalos, P.M.: The hot-rolling batch scheduling method based on the prize collecting vehicle routing problem. J. Ind. Manag. Optim. **5**(4), 749–765 (2009)

A Time-Indexed Generalized Vehicle Routing Model and Stabilized Column Generation for Military Aircraft Mission Planning

Nils-Hassan Quttineh, Torbjörn Larsson, Jorne Van den Bergh, and Jeroen Beliën

Abstract We introduce a time-indexed mixed-integer linear programming model for a military aircraft mission planning problem, where a fleet of cooperating aircraft should attack a number of ground targets so that the total expected effect is maximized. The model is a rich vehicle routing problem and the direct application of a general solver is practical only for scenarios of very moderate sizes. We propose a Dantzig–Wolfe reformulation and column generation approach. A column here represents a specific sequence of tasks at certain times for an aircraft, and to generate columns a longest path problem with side constraints is solved. We compare the column generation approach with the time-indexed model with respect to upper bounding quality of their linear programming relaxations and conclude that the former provides a much stronger formulation of the problem.

Keywords Aircraft mission planning problem • Time-indexed mixed-integer linear program • Vehicle routing problem • Dantzig–Wolfe method • Column generation

1 Introduction

We study a military aircraft mission planning problem (MAMPP), which was introduced by Quttineh et al. [27]. In general, a military aircraft mission might involve various tasks, such as surveillance, backup support, rescue assistance, or an attack. We only consider the situation where a set of ground targets needs to be

N.-H. Quttineh (✉) • T. Larsson
Linköping University, 58183 Linköping, Sweden
e-mail: nils-hassan.quttineh@liu.se

J. Van den Bergh • J. Beliën
KU Leuven, Campus Brussels, Warmoesberg 26, 1000 Brussels, Belgium

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_15

attacked with a fleet of aircraft. The planning of such aircraft missions is still to a large extent carried out manually, and it takes an experienced planner several hours to create a feasible plan.

The research presented here has been performed in collaboration with an industrial partner and is a continuation of the work by Quttineh et al. [25–27]. The MAMPP is recognized as a generalized vehicle routing problem (GVRP) with precedence relationships and synchronization in time and position between multiple vehicles. Examples of mathematical optimization approaches to military routing problems can be found in [7, 29, 30, 34]. To the best of our knowledge, the MAMPP has not been analyzed by optimization methods by others.

Synchronization in a vehicle routing problem (VRP) might be exhibited with regard to spatial, temporal, and load aspects. A recent survey of VRPs with synchronization constraints (VRPS) is given in Drexl [10] and shows that this topic is challenging and emerging. Following the definitions from this paper, the synchronization in our problem can be classified as operation synchronization, in which one has to decide about time and location of some interaction between vehicles. In [11], Drexl presents modeling techniques for a VRP with trailers and transshipments (VRPTT), which is an application of the VRP with all the previously mentioned synchronization constraints. Different transformations of classic VRPs and of several types of VRPS are described. Recently, Drexl [12] presented two mixed-integer programming formulations and five branch-and-cut algorithms for the VRPTT.

Bredström and Rönnqvist [6] give a daily homecare planning problem, which is modeled as a vehicle routing and scheduling problem with precedence constraints on visits as well as time windows and pairwise synchronization (because two staff members are required to visit an elderly person simultaneously). Redjem et al. [28] also consider routing with time windows and synchronized visits for a homecare planning problem. Synchronized routing and scheduling problems need to be solved also in the forestry industry. El Hachemi et al. [14], for instance, include multiple aspects such as pickup and delivery, and inventory stock, and solve the decomposed problem using constraint-based local search. Other examples of work on routing with synchronization are [1, 3, 21].

Already in the 1970s, Golden [17] touched the GVRP as a variation of the classic VRP. One of the first dedicated papers on GVRP is by Ghiani and Improta [16], who give a transformation to the capacitated arc routing problem. Baldacci et al. [4] discuss some applications for the GVRP, whereas formulations and branch-and-cut algorithms are given in the recent paper of Bektaş et al. [5]. Hà et al. [18] solve the GVRP with the number of vehicles as a decision variable, both heuristically and exact using a branch-and-cut approach. For the same problem, Afsar et al. [2] present an exact method based on column generation, and two metaheuristics.

In Sigurd et al. [31], vehicle routing with precedence constraints and time windows is considered in order to schedule transportation of live animals to avoid the spread of diseases. A general framework for VRP with time windows and temporal dependencies, including exact synchronization, is given in Dohn et al. [9]. In the context of GVRP, a time windows extension is considered by Moccia

et al. [24], who suggest a metaheuristic solution method. Their work concerns an application to the design of home-to-work transportation plans.

By taking into account multiple non-standard characteristics of the GVRP, such as precedence relationships and operation synchronization, we believe to contribute to the existing literature. Our chapter reads as follows. In Sect. 2, the problem setting is described, followed by a time-indexed mathematical formulation in Sect. 3. Section 4 develops a column generation method for a Dantzig–Wolfe reformulation of the time-indexed model, followed in Sect. 5 by a description of a stabilized column generation method. In Sect. 6, we give theoretical bounding results. Further, in Sect. 7, numerical results of our approach are discussed, followed by a conclusion in Sect. 8.

2 Problem Setting

This section provides a concise description of the problem setting. A detailed report on the complex problem characteristics and how to transform them into a mathematical formulation can be found in Quttineh et al. [27]. As mentioned above, we only consider military aircraft missions involving attacks. The geographical area of interest, referred to as the target scene, includes the targets that need to be attacked and other objects such as enemy defense positions, like surface-to-air missiles (SAMs), and protected objects, like hospitals and schools. We consider all objects to be stationary with known positions. The target scene is defined by a line of entrance and a line of exit for the aircraft. These are typically deployed from a base situated far away from the target scene and enter the scene by the entry line, carry out the mission and return to a base after leaving the scene at the exit line. The diameter of a target scene is usually of the order of 100 km, the distances between targets are of the order of a few kilometers, and the time span of the attacks is around a quarter of an hour. Typically, a mission involves 6–8 targets and 4–6 aircraft. At the end of this section, an example of a target scene is depicted, together with a solution.

The goal of a mission is to find an attack plan where maximal total expected effect is gained within short time span. The mission time is defined by the time the first aircraft passes the entry line and the time the last aircraft passes the exit line. Since the entire target scene is located in hostile area, the mission time needs to be minimized. To take into account the threat from defense positions, aircraft are restricted not to fly through defended airspace. Weapons, on the other hand, are allowed to pass through defended airspace, but at the risk of being shot down, that is, with a lower expected effect on the target.

In order to plan a mission, the aircraft characteristics need to be taken into account. Each aircraft has an armament capacity, limiting the number of attacks it can perform. It can also be equipped with an illumination laser pod to guide weapons. Each target needs to be attacked exactly once and requires one aircraft that illuminates the target with a laser beam and one aircraft that launches the weapon. Since an attack requires continuous illumination from the launch of the weapon until

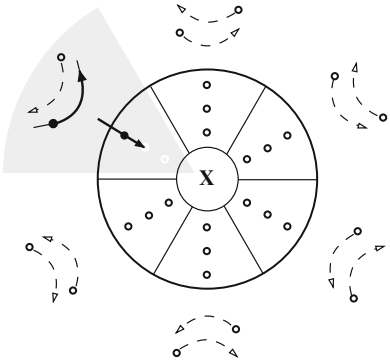


Fig. 1 The feasible attack space defined by inner and outer radii and divided into six sectors, each with three attack and two illumination alternatives. A pair of compatible attack and illumination positions is marked, where the *arrows* indicate the flight directions

its impact, the two aircraft need to team up. This rendezvous not only depends on the time but also on the location of both aircraft, so that the illumination is continuously visible for the weapon.

Figure 1 illustrates how a target is modeled. The feasible attack space can be derived from the type of aircraft and the type of weapon being used and is represented by the inner and outer radii. This attack space is then divided into six sectors, which each holds at most three discretized attack positions and two compatible illumination positions. If a protected object is inside the estimated area of risk for collateral damage of a given attack position, this position is considered unfeasible. For any attack position, the expected effect on the target can be calculated. It depends on the kind of weapon being used, which is decided in advance, and on the direction of the impact and the weapon's kinetic energy. The two illumination alternatives per sector differ in flight direction, roughly clockwise or counterclockwise, but are both compatible with all attack positions of the sector. In our problem setting we consider only one altitude layer, but one could of course extend the target modeling by allowing attack options on different discrete altitude layers.

Not all attack sequences are allowed. Depending on the wind direction and the proximity between targets, dust and debris might reduce the visibility and hinder an attack. Hence, we assume that precedence constraints are given, specifying which targets are not allowed to be attacked before other targets.

In summary, the problem involves three types of decisions. First, the choice of attack direction against each target. Second, which two aircraft shall be assigned against the targets. Third, the order in which each aircraft fulfils its assigned tasks in the mission. Now it is clear that the problem belongs to the class of VRPs, describing the attack and illumination positions by nodes, each of which being associated with an expected effect on the target. By further introducing dummy nodes associated with the crossings of the entry and exit lines of the target scene, and modeling possible aircraft movements by arcs, the mission planning problem can partly be

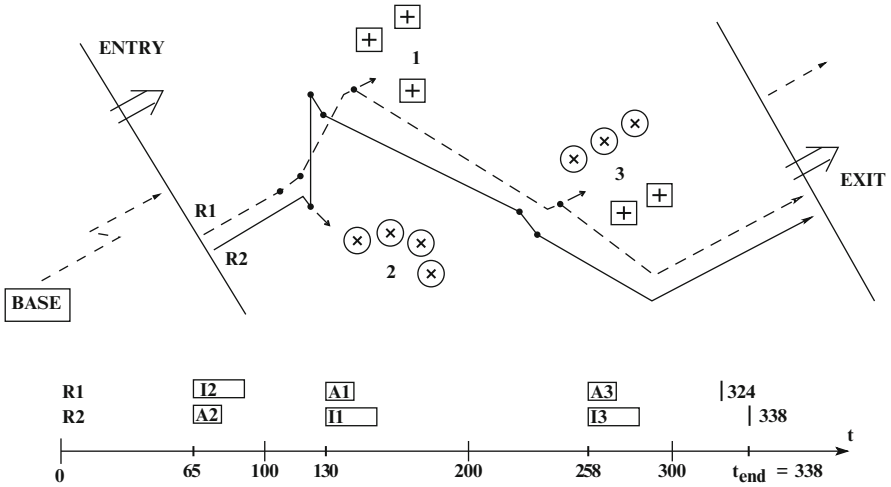


Fig. 2 Optimal solution to a problem instance that includes three targets and nearby SAMs (*times symbol*) and hospitals (*plus symbol*). Shown are aircraft routes, chosen attack and illumination positions against each target, the times of the attacks, and the times when the two aircraft pass the exit line

represented by a network. Because of the precedence relationships, some arcs are eliminated from the network. The restriction that every target should be attacked exactly once results in a network that only contains arcs between different targets, or from or to the dummy nodes.

Each of the arcs has two attributes: an expected effect and a travel time. The effect attribute is different from zero only for an arc that is leaving an attack node, and it then equals the resulting expected effect against the target. A flight path between two positions has to comply with restrictions on the aircraft dynamics and that the aircraft cannot pass through defended airspace. By using a flight path generator provided by our industrial partner, we are able to find the path with minimal time between any pair of positions. In general, travel times will be asymmetric because each position is also associated with a flight direction.

To illustrate the essential aspects of a solution to the MAMPP, Fig. 2 depicts a target scene and an optimal solution. For this problem instance, two aircraft are used, there are no precedence constraints on the targets, and each aircraft can attack at most two targets. All numerical data used in the scenario were provided by our industrial partner.

The aircraft routes are shown as solid and dashed lines. The attack sequence is 2–1–3, with a total mission time of $t_{end} = 338$ s. The expected effects of the attacks on targets 2 and 3 are maximal, among the available attack positions for these targets, while the attack position against target 1 renders an effect that is slightly below the maximal possible. Achieving maximal effect against this target would require a longer tour for both aircraft, which makes this alternative nonoptimal.

3 A Time-Indexed Mathematical Model

We here present a time-indexed mixed-integer linear programming (MILP) mathematical model of the MAMPP. This MILP model can be derived from the one introduced by Quttineh et al. [27], through a discretization of time. In particular, this discretization allows an alternative modeling of the time propagation constraints. We divide the nomenclature into indices and sets, parameters and coefficients, and decision variables, given in Tables 1, 2, and 3.

The primary objective is to maximize the total expected effect against all the targets. However, in order to achieve this effect, the use of long flight paths within the target scene might be necessary, which exposes the aircraft to a higher risk of being detected and engaged by enemy defense. A secondary objective is therefore to limit the mission time span. We thus have a multi-objective optimization problem, with two objectives that are typically in conflict.

Table 1 Indices and sets

$\mathbf{R}$	Fleet of aircraft, r
$\mathbf{M}$	Set of targets, m, to be attacked
$\mathbf{N}$	Set of nodes in the network, excluding the origin (o) and destination (d) nodes
$\mathbf{G}, \mathbf{G}_m$	Set of all sectors for all targets and for target m, respectively
$\mathbf{N}_m^A, \mathbf{N}_m^I$	Set of feasible attack (A) and illumination (I) nodes, respectively, for target m
$\mathbf{A}, \mathbf{A}_g, \mathbf{I}_g$	Set of arcs in the network (including from o and to d) and sets of arcs (i,j) such that node j is an attack (A) node or illumination (I) node in sector g, respectively
$\mathbf{P}$	Set of ordered pairs (m,n) of targets such that the attack on target m cannot precede the attack on target n
$\mathbf{S}$	Set of time periods within a discretized planning horizon, each of step length Δt

Table 2 Parameters

c_{ij}^r	For arcs (i,j) with $i \in N_m^A$, that is, for arcs leaving attack nodes, the value of c_{ij}^r is the expected effect of the attack, and otherwise the value is zero
S_{ij}^r	The time needed for aircraft r to traverse arc (i,j), expressed in number of time periods; equals actual time to traverse the arc divided by Δt, rounded upwards
T_s	The ending time of period s, which equals $s \cdot \Delta t$, $s = 0, 1, \ldots, \lvert\mathbf{S}\rvert$
Γ^r	Armament capacity of aircraft r
q_m	Weapon capacity needed towards target m
μ	Positive parameter that weights mission time span against expected effect on targets

Table 3 Decision variables

x_{ij}^r	Routing variable, equals 1 if aircraft r traverses arc (i,j) and 0 otherwise
y_{is}^r	Time indicator variable, equals 1 if node i is visited by aircraft r in time period s and 0 otherwise
t_{end}	The time that the last aircraft passes the exit line

Since the maximal allowed mission time span is given by $|\mathbf{S}| \cdot \Delta t$, an explicit way of limiting the mission time span is to reduce the cardinality of $\mathbf{S}$, which might however cause the MAMPP to become infeasible. A further drawback of this approach is that it can allow mission time spans that are unnecessarily long with respect to the obtained target effect.

Instead, we have chosen to optimize a weighted combination of the two objectives, using the positive parameter μ which reflects the trade-off between effect on target and mission time span. This yields a solution that is Pareto optimal. As part of a decision support tool, the value of μ can be either chosen by a mission planner or varied systematically in order to generate a population of mission plans with different properties with respect to effect and time, to be further evaluated by a mission planner. Since target effect is the primary goal, the value of μ is typically small.

The time-indexed mathematical model for the MAMPP is given below:

$$z^*_{IP} = \max \sum_{r\in\mathbf{R}} \sum_{(i,j)\in\mathbf{A}} c^r_{ij} x^r_{ij} - \mu t_{end} \qquad \text{[TI-MAMPP]}$$

subject to

$$\sum_{(o,j)\in\mathbf{A}} x^r_{oj} = 1, \qquad r \in \mathbf{R} \qquad (1)$$

$$\sum_{(i,d)\in\mathbf{A}} x^r_{id} = 1, \qquad r \in \mathbf{R} \qquad (2)$$

$$\sum_{(i,k)\in\mathbf{A}} x^r_{ik} = \sum_{(k,j)\in\mathbf{A}} x^r_{kj}, \qquad k \in \mathbf{N},\ r \in \mathbf{R} \qquad (3)$$

$$\sum_{r\in\mathbf{R}} \sum_{g\in\mathbf{G}_m} \sum_{(i,j)\in\mathbf{A}_g} x^r_{ij} = 1, \qquad m \in \mathbf{M} \qquad (4)$$

$$\sum_{r\in\mathbf{R}} \sum_{g\in\mathbf{G}_m} \sum_{(i,j)\in\mathbf{I}_g} x^r_{ij} = 1, \qquad m \in \mathbf{M} \qquad (5)$$

$$\sum_{r\in\mathbf{R}} \sum_{(i,j)\in\mathbf{A}_g} x^r_{ij} = \sum_{r\in\mathbf{R}} \sum_{(i,j)\in\mathbf{I}_g} x^r_{ij}, \qquad g \in \mathbf{G} \qquad (6)$$

$$\sum_{g\in\mathbf{G}_m} \sum_{(i,j)\in\mathbf{A}_g\cup\mathbf{I}_g} x^r_{ij} \le 1, \qquad m \in \mathbf{M},\ r \in \mathbf{R} \qquad (7)$$

$$\sum_{m\in\mathbf{M}} \sum_{g\in\mathbf{G}_m} \sum_{(i,j)\in\mathbf{A}_g} q_m x^r_{ij} \le \Gamma^r, \qquad r \in \mathbf{R} \qquad (8)$$

$$y^r_{o0} = 1, \qquad r \in \mathbf{R} \qquad (9)$$

$$\sum_{t=s+S^r_{ij}}^{|\mathbf{S}|} y^r_{jt} \ge x^r_{ij} + y^r_{is} - 1, \qquad (i,j) \in \mathbf{A},\ s \in \{0\}\cup\mathbf{S},\ r \in \mathbf{R} \qquad (10)$$

$$\sum_{s\in\mathbf{S}} y^r_{ks} = \sum_{(k,j)\in\mathbf{A}} x^r_{kj}, \qquad k\in\mathbf{N},\ r\in\mathbf{R} \tag{11}$$

$$\sum_{r\in\mathbf{R}}\sum_{i\in\mathbf{N}^A_m} y^r_{is} = \sum_{r\in\mathbf{R}}\sum_{i\in\mathbf{N}^I_m} y^r_{is}, \qquad m\in\mathbf{M},\ s\in\mathbf{S} \tag{12}$$

$$\sum_{r\in\mathbf{R}}\sum_{t=s}^{|\mathbf{S}|}\sum_{i\in\mathbf{N}^A_m} y^r_{it} \geq \sum_{r\in\mathbf{R}}\sum_{i\in\mathbf{N}^A_n} y^r_{is}, \qquad (m,n)\in\mathbf{P},\ s\in\mathbf{S} \tag{13}$$

$$\sum_{i\in\mathbf{N}^A_m}\sum_{t=1}^{s-1} y^r_{it} + \sum_{i\in\mathbf{N}^A_n} y^r_{is} \leq 1, \qquad (m,n)\in\mathbf{P},\ s\in\mathbf{S},\ r\in\mathbf{R} \tag{14}$$

$$\sum_{s\in\mathbf{S}} y^r_{is} \leq 1, \qquad i\in\mathbf{N}\cup\{o,d\},\ r\in\mathbf{R} \tag{15}$$

$$\sum_{s\in\{0\}\cup\mathbf{S}} T_s y^r_{ds} \leq t_{end}, \qquad r\in\mathbf{R} \tag{16}$$

$$x^r_{ij}\in\{0,1\}, \qquad (i,j)\in\mathbf{A},\ r\in\mathbf{R} \tag{17}$$

$$y^r_{is}\in\{0,1\}, \qquad i\in\mathbf{N}\cup\{o,d\},\ s\in\{0\}\cup\mathbf{S},\ r\in\mathbf{R} \tag{18}$$

Constraints (1) and (2) describe that each aircraft leaves and enters the target scene via the origin and destination nodes, respectively, while constraint (3) is the node balance equation for each aircraft. The requirement that each target shall be attacked and illuminated exactly once is modeled by constraints (4) and (5), respectively, while constraint (6) synchronizes these tasks to the same sector. Constraint (7) states that each aircraft can visit each target at most once. This constraint is actually redundant, but it strengthens the column generation problems to be presented. The armament limitation is modeled by constraint (8).

Further, constraint (9) states that each aircraft is leaving the origin at time zero. Constraint (10) ensures that if aircraft r is visiting node j directly after node i, then the time of visiting node j cannot be earlier than the time of visiting node i plus the time needed to traverse arc (i,j). Constraint (11) enforces that if node i is not visited by an aircraft, no outgoing arc (i,j) from that node can be traversed by the aircraft.

Constraint (12) states that the attack and the illumination of a target need to be synchronized in time. Constraint (13) imposes the precedence restrictions on the attacking times of pairs of targets. Similarly, constraint (14) imposes the precedence restrictions for an individual aircraft. This constraint is also redundant, but it strengthens the column generation problems. Constraint (15) states that each

aircraft can visit each node in at most one time period, and constraint (16) defines the total mission time, since all aircraft end up at the destination node. Finally, (17) and (18) are definitional constraints.

The optimal value of the linear programming (LP) relaxation of TI-MAMPP is denoted z^*_{LP}.

4 Column Generation

The planning of a military aircraft mission is typically made close to when the mission actually takes place (say, within 24 h); one reason for this is that the planning can then be based on the most recent information. The time needed for the chain of planning is of the order of several hours. Solving the continuous time version of MAMPP presented in Quttineh et al. [27] to optimality takes a general MIP solver several hours for already moderate-sized problem instances. This is also the case for the model TI-MAMPP presented above. Hence, efficient algorithms are needed to meet the needs and expectations in a real-life setting. We propose a column generation method based on a Dantzig–Wolfe reformulation [8] of the model TI-MAMPP. For overviews of column generation, see, for example, [22] and [33].

The Dantzig–Wolfe reformulation is defined in the following steps. Suppose that constraints (1)–(3), (7)–(11), (14)–(15), and (17)–(18) have N_r feasible solutions for aircraft $r \in \mathbf{R}$. Each of these describes a possible route for the aircraft, involving specific tasks at specific targets at certain times. Assume that $n_r < N_r$ of the routes for aircraft $r \in \mathbf{R}$ is explicitly available. Typically, $n_r \ll N_r$ holds. Let the values of the variables for each feasible solution to the above-mentioned constraints be denoted by x^{rk}_{ij} and y^{rk}_{is}, $k = 1, \ldots, n_r$.

Next, we relax the binary variable restrictions from the TI-MAMPP and introduce variables z^r_k as convexity weights on the solutions x^{rk}_{ij} and y^{rk}_{is}, $k = 1, \ldots, n_r$. Further, we impose the relationships

$$x^r_{ij} = \sum_{k=1}^{n_r} x^{rk}_{ij} z^r_k \quad \text{and} \quad y^r_{is} = \sum_{k=1}^{n_r} y^{rk}_{is} z^r_k.$$

Substitution of these relationships into the objective function and into constraints (4)–(6), (12), (13), and (16) yields the following restricted Dantzig–Wolfe master problem:

$$z^*_{RMP} = \max \sum_{r \in \mathbf{R}} \sum_{k=1}^{n_r} \left(\sum_{(i,j) \in \mathbf{A}} c^r_{ij} x^{rk}_{ij} \right) z^r_k - \mu t_{end} \qquad \text{[DW–RMP]}$$

subject to

$$[\alpha_m] \quad \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{g\in\mathbf{G}_m}\sum_{(i,j)\in\mathbf{A}_g} x_{ij}^{rk}\right) z_k^r = 1, \qquad m\in\mathbf{M} \qquad (19)$$

$$[\beta_m] \quad \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{g\in\mathbf{G}_m}\sum_{(i,j)\in\mathbf{I}_g} x_{ij}^{rk}\right) z_k^r = 1, \qquad m\in\mathbf{M} \qquad (20)$$

$$[\gamma_g] \quad \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{(i,j)\in\mathbf{A}_g} x_{ij}^{rk}\right) z_k^r = \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{(i,j)\in\mathbf{I}_g} x_{ij}^{rk}\right) z_k^r, \qquad g\in\mathbf{G} \qquad (21)$$

$$[\eta_{ms}] \quad \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{i\in\mathbf{N}_m^A} y_{is}^{rk}\right) z_k^r = \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{i\in\mathbf{N}_m^I} y_{is}^{rk}\right) z_k^r, \qquad m\in\mathbf{M},\ s\in\mathbf{S} \qquad (22)$$

$$[\lambda_{mns}] \quad \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{t=s}^{|\mathbf{S}|}\sum_{i\in\mathbf{N}_m^A} y_{it}^{rk}\right) z_k^r \geq \sum_{r\in\mathbf{R}}\sum_{k=1}^{n_r}\left(\sum_{i\in\mathbf{N}_n^A} y_{is}^{rk}\right) z_k^r, \qquad s\in\mathbf{S},\ (m,n)\in\mathbf{P} \qquad (23)$$

$$[\tau_r] \quad \sum_{k=1}^{n_r}\left(\sum_{s\in\{0\}\cup\mathbf{S}} T_s\cdot y_{ds}^{rk}\right) z_k^r \leq t_{end}, \qquad r\in\mathbf{R} \qquad (24)$$

$$[\nu_r] \quad \sum_{k=1}^{n_r} z_k^r = 1, \qquad r\in\mathbf{R} \qquad (25)$$

$$z_k^r \geq 0, \qquad k=1,\ldots,n_r,\ r\in\mathbf{R} \qquad (26)$$

Each column of this problem represents a route for a specific aircraft, and the restricted master problem is to find the best way to combine all available routes into a solution that is feasible and optimal with respect to the restrictions that couple all aircraft, in a linear programming sense.

Comparing DW–RMP with TI-MAMPP, constraints (19)–(21) correspond to the attack, illumination, and synchronization constraints (4)–(6), while constraints (22) and (23) match the time synchronization and precedence constraints (12) and (13). Further, constraint (24) defines the total mission time, similarly to (16). Finally, constraints (25) and (26) are definitional.

If all feasible routes for each aircraft are known, that is, if $n_r = N_r$ holds for all $r\in\mathbf{R}$, the restricted master problem becomes a full master problem, with an optimal objective value denoted z_{MP}^*. Further, any optimal solution to DW–RMP that is integral yields a feasible solution to TI-MAMPP and a lower bound to z_{IP}^*, denoted $\underline{z}_{IP}$.

Assume that DW–RMP has a feasible solution. Each of its constraints is associated with a dual variable, indicated in the square brackets to the left. The optimal values of these dual variables are used to define a Dantzig–Wolfe subproblem, or column generation problem, for each aircraft $r \in \mathbf{R}$. The objective function in each subproblem describes the reduced cost of any feasible column, that is, any possible route for the aircraft. As long as there is a route with a positive reduced cost, such routes should be generated and their corresponding columns added to DW–RMP. Generating columns with positive reduced costs boils down to solving the following subproblem for each aircraft $r \in \mathbf{R}$:

$$\begin{aligned}
\bar{c}^{r}_{n_r+1} \;=\; \max \sum_{(i,j)\in\mathbf{A}} c^{r}_{ij}x^{r}_{ij} - \tau_r \sum_{s\in\{0\}\cup\mathbf{S}} T_s y^{r}_{ds} - \qquad & [\text{DW–SUB}_r] \\
- \sum_{m\in\mathbf{M}} \left(\alpha_m \sum_{g\in\mathbf{G}_m} \sum_{(i,j)\in\mathbf{A}_g} x^{r}_{ij} - \beta_m \sum_{g\in\mathbf{G}_m} \sum_{(i,j)\in\mathbf{I}_g} x^{r}_{ij} \right) - & \\
- \sum_{g\in\mathbf{G}} \gamma_g \left(\sum_{(i,j)\in\mathbf{A}_g} x^{r}_{ij} - \sum_{(i,j)\in\mathbf{I}_g} x^{r}_{ij} \right) & \\
- \sum_{m\in\mathbf{M}} \sum_{s\in\mathbf{S}} \eta_{ms} \left(\sum_{i\in\mathbf{N}^A_m} y^{r}_{is} - \sum_{i\in\mathbf{N}^I_m} y^{r}_{is} \right) - & \\
- \sum_{(m,n)\in\mathbf{P}} \sum_{s\in\mathbf{S}} \lambda_{mns} \left(\sum_{t=s}^{|\mathbf{S}|} \sum_{i\in\mathbf{N}^A_m} y^{r}_{it} - \sum_{i\in\mathbf{N}^A_n} y^{r}_{is} \right) - \nu_r &
\end{aligned}$$

subject to $(1), (2), (3), (7), (8), (9), (10), (11), (14), (15), (17), (18)$.

The problem DW–SUB$_r$ can be described as a side constrained longest path problem in a time-layered network where all nodes in $\mathbf{N}$ have $|\mathbf{S}|$ time copies. Constraints (9)–(11) are taken into account implicitly in the construction of the network, while constraints (7), (8), (14), and (15) are side constraints. This problem does not possess the integrality property.

An upper bound on z^*_{MP} is given by $z^*_{RMP} + \sum_{r\in\mathbf{R}} \bar{c}^{r}_{n_r+1}$. The lowest such upper bound ever found is denoted by $\bar{z}_{MP}$.

5 Stabilized Column Generation

As is well known, column generation methods are dually equivalent to cutting plane methods. The latter are known to be inherently instable [19] in the sense that successive iterates can be very distant, which may cause slow convergence.

In column generation methods, the dual instability manifests itself as oscillations in the values of the dual variables, which slows down the convergence also in the primal space.

In order to improve the efficiency of the column generation scheme, it is therefore common to apply a stabilization of the values of the dual variables. This technique was introduced by Marsten et al. [23] back in 1975, and examples of applications from more recent years can be found in [13] and [32], to mention some.

The idea is to prevent the dual solution of the DW–RMP to fluctuate between successive iterations. This is accomplished by including a box constraint for each dual variable, centered around its current value and preventing the value to change drastically from one iteration to the next. These additional constraints in the dual problem correspond to auxiliary variables in the primal problem, and the effect of these variables is a relaxation of the original primal constraints. Consequently, the parameters that specify the size of the box appear as penalty weights in the objective function for the auxiliary variables.

We stabilize constraints (19)–(23) in DW–RMP, and the optimal objective value of the stabilized DW–RMP is denoted z^*_{SRMP}. An upper bound on z^*_{MP} is calculated as $z^*_{SRMP} + \sum_{r\in\mathbf{R}} \bar{c}^r_{n_r+1}$. (The reason that a formula similar to the one used in non-stabilized column generation applies also in the stabilized case is that both formulas are in fact equivalent to a Lagrangian dual bound and that it is of no significance how the dual point is obtained.)

The size of each box slowly shrinks every iteration, and it is re-centered every time it becomes binding (that is, every time an auxiliary variable becomes nonzero).

6 Bounding Properties

The relationships between the various optimal values and bounds in our column generation approach become rather intricate. These relationships are illustrated in Fig. 3.

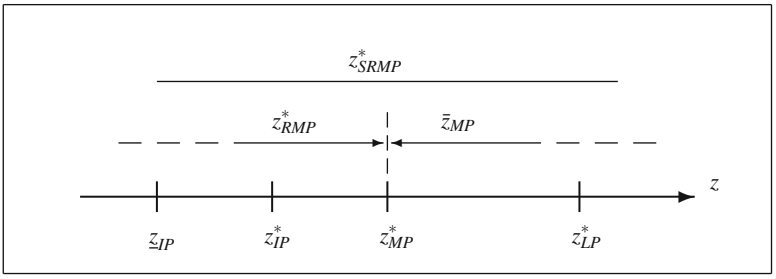


Fig. 3 Bounding relationships for the column generation approach

The optimal value z^*_{IP} for TI-MAMPP is trivially bounded from below by the objective value, z_{IP}, of any feasible solution, and bounded above by the optimal LP value z^*_{LP}. This bound has proven to be very weak, see [25, 27]. Further, z^*_{IP} can be bounded from above by the optimal LP value of the full master problem, z^*_{MP}. It always holds that $z^*_{MP} \leq z^*_{LP}$, but since the column generation problem DW–SUB$_r$ does not have the integrality property, $z^*_{MP} < z^*_{LP}$ can be expected to hold.

Assume first that no stabilization is used. As routes are added to the restricted master problem, its optimal value z^*_{RMP} converges monotonically towards z^*_{MP}. Note that the relationship between z^*_{RMP} and z^*_{IP} is unknown. Further, $\bar{z}_{MP}$ is convergent towards z^*_{MP} from above.

Considering the case with stabilization, the relationship between z^*_{SRMP} and z^*_{MP} is unknown, since the stabilized restricted master problem includes both a restriction and a relaxation, as compared to the full master problem. However, the value z^*_{SRMP} becomes a lower bound for z^*_{MP} if the dual box is not binding (that is, all auxiliary variables in the primal problem are zero). Finally, $\bar{z}_{MP}$, as calculated in Sect. 5, is an upper bound for z^*_{MP}. Further, it converges towards z^*_{MP}.

7 Numerical Validation

We have made a preliminary assessment of TI-MAMPP and the column generation approach by using a few small problem instances that are identical to, or slight modifications of, instances used in [27]. All experiments have been carried out using the modeling language AMPL [15] and the solver CPLEX [20].

Table 4 shows problem characteristics and results obtained with the continuous-time model of MAMPP in [27] and TI-MAMPP. We observe that even for rather large time steps, the optimal solutions found by the continuous-time and time-indexed models are very similar, with respect to attack sequences and to attack and illumination nodes. Although not reported in the table, we also observe that the solution times of the continuous-time and time-indexed models are similar for

Table 4 Problem characteristics and comparison of the continuous-time and time-indexed models

	Problem			Cont.		$\Delta t = 60$		$\Delta t = 45$		$\Delta t = 30$	
No.	$\|\mathbf{M}\|$	prec.	Γ^r	Eff.	t_{end}	Eff.	t_{end}	Eff.	t_{end}	Eff.	t_{end}
1	3	–	3	0.974	333	0.808	420	0.974	405	0.974	390
2	3	–	2	0.974	338	0.808	420	0.974	405	0.974	390
3	3	$\{1\|23\}$	3	0.863	352	0.808	420	0.863	405	0.808	390
4	4	$\{1\|2\|3\|4\}$	3	0.917	628	1.000	840	0.917	720	0.917	720
5	4	$\{1\|2\|3\|4\}$	2	0.917	638	1.000	840	0.917	720	0.917	720

Here, $\mu = 0.005$ and all instances include two aircraft. The notation $\{1|23\}$ means that target 1 is attacked before targets 2 and 3. The maximal possible total effect on targets is 1.000

Table 5 Comparison of the time-indexed model and column generation

	Time-indexed			CG: $\Delta t = 45$			CG: $\Delta t = 30$		
No.	z^*_{LP}	$z^*_{IP}[45]$	$z^*_{IP}[30]$	z^*_{MP}	$\underline{z}_{IP}$	Iter.	z^*_{MP}	$\underline{z}_{IP}$	Iter.
1	23.173	1.933	2.683	1.933	1.933	16	2.683	2.683	22
2	23.173	1.887	2.674	1.887	1.887	11	2.674	2.674	15
3	22.813	0.346	0.080	0.346	0.346	22	1.271	–	22
4	30.117	−7.677	−7.730	−6.532	–	37	−4.744	–	37
5	30.115	−7.730	−7.730	−7.083	–	29	−6.002	–	60

The optimal LP value z^*_{LP} of the time-indexed model varies very little with the step size; we give the value for $\Delta t = 60$. The columns $z^*_{IP}[45]$ and $z^*_{IP}[30]$ are the optimal values of the time-indexed model with different time steps. Further, $\underline{z}_{IP}$ are the objective values obtained when solving the integer version of the final master problem (and a feasible solution exists), and Iter. is the number of column generation iterations needed to reach optimality

large time steps, while the latter is much more demanding when the steps are small. Further, the upper bounds given by the linear programming relaxations of the continuous-time and time-indexed versions of the MAMPP are very similar, independent of the sizes of the time steps, and very weak.

Table 5 shows a comparison between the time-indexed model and the column generation approach. Here, initial values for the dual variables for the stabilized constraints (19)–(23), used to initialize the dual boxes, are obtained by solving the LP relaxation of TI-MAMPP. (The radii of the boxes were initially set to 0.3 and shrinked by a factor of 0.97 in each iteration.) To create an initial set of routes and columns, the DW–SUB$_r$ problem is solved for an ad hoc fixed set of sectors to be visited, for each aircraft $r \in \mathbf{R}$.

Comparing the columns z^*_{LP} and z^*_{MP} with the columns z^*_{IP}, we conclude that the upper bound on z^*_{IP} obtained from z^*_{MP} is much tighter than the bound z^*_{LP}. The bound z^*_{MP} is indeed very close to z^*_{IP} while the bound z^*_{LP} is very weak. Further, comparing the columns z^*_{IP} and $\underline{z}_{IP}$, we see that whenever the restricted master problem has an integral feasible solution, it is also of high quality.

8 Conclusion

Clearly, the Dantzig–Wolfe reformulation and column generation approach provide vastly superior upper bounds on the optimal value of TI-MAMPP. We conclude that the Dantzig–Wolfe reformulation gives rise to a very strong formulation of the TI-MAMPP. This model by itself is not very efficient in terms of solving the military aircraft mission planning problem, but it was helpful in the development of the column generation procedure.

The solution times of our implementation of the column generation approach are not competitive compared to direct methods. The solution of DW–RMP takes

very little time. This holds even for the integer version of this problem. The column generation problem DW–SUB$_r$ is however very time-consuming to solve to optimality.

There are several opportunities for tailoring and streamlining the computations, and especially to reduce the computational burden of the column generation problem DW–SUB$_r$. For example, in early column generation iterations it might be more efficient to terminate the column generation solver as soon as the objective value gets positive, since this is enough to ensure progress. Further, a tailored solver for DW–SUB$_r$ can be developed by exploiting its underlying time-layered network structure. This is an interesting opportunity for further research.

The column generation approach can be applied to obtain an upper bound, to be used for assessing the quality of any feasible solution to TI-MAMPP, for example, generated by a metaheuristic. Also, feasible solutions generated by metaheuristics can be used to provide high quality initial columns to the restricted master problem. This combination is another topic for further research.

A great advantage of the column generation approach to MAMPP in a real-life planning situation would be its creation of many possible routes for all aircraft. This is of practical interest since a real-life MAMPP can never be expected to include all possible aspects of the mission to be planned, and because of the multi-objective nature of the problem. The access to multiple aircraft routes can then be exploited in an interactive decision support system.

References

1. Afifi, S., Dang, D.-C., Moukrim, A.: A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints. In: Nicosia, G., Pardalos, P. (eds.) Learning and Intelligent Optimization. Lecture Notes in Computer Science, pp. 259–265. Springer, Berlin (2013)
2. Afsar, H.M., Prins, C., Santos, A.C.: Exact and heuristic algorithms for solving the generalized vehicle routing problem with flexible fleet size. Int. Trans. Oper. Res. **21**, 153–175 (2014)
3. Andersson, H., Duesund, J.M., Fagerholt, K.: Ship routing and scheduling with cargo coupling and synchronization constraints. Comput. Ind. Eng. **61**, 1107–1116 (2011)
4. Baldacci, R., Bartolini, E., Laporte, G.: Some applications of the generalized vehicle routing problem. J. Oper. Res. Soc. **61**, 1072–1077 (2010)
5. Bektaş, T., Erdoğan, G., Røpke, S.: Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. Transp. Sci. **45**, 299–316 (2011)
6. Bredström, D., Rönnqvist, M.: Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. Eur. J. Oper. Res. **191**, 19–31 (2008)
7. Carlyle, W.M., Royset, J.O., Wood, R.K.: Lagrangian relaxation and enumeration for solving constrained shortest-path problems. Networks **52**, 256–270 (2008)
8. Dantzig, G.B., Wolfe, P.: Decomposition principles for linear programs. Oper. Res. **8**, 101–111 (1960)
9. Dohn, A., Rasmussen, M.S., Larsen, J.: The vehicle routing problem with time windows and temporal dependencies. Networks **58**, 273–289 (2011)
10. Drexl, M.: Synchronization in vehicle routing – a survey of VRPs with multiple synchronization constraints. Transp. Sci. **46**, 297–316 (2012)

11. Drexl, M.: Applications of the vehicle routing problem with trailers and transshipments. Eur. J. Oper. Res. **227**, 275 –283 (2013)
12. Drexl, M.: Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. Networks **63**, 119–133 (2014)
13. du Merle, O., Villeneuve, D., Desrosiers, J., Hansen, P.: Stabilized column generation. Discret. Math. **194**, 229–237 (1999)
14. El Hachemi, N., Gendreau, M., Rousseau, L.M.: A heuristic to solve the synchronized log-truck scheduling problem. Comput. Oper. Res. **40**, 666–673 (2014)
15. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL – A Modeling Language for Mathematical Programming. Duxbury Press, Belmont (2003)
16. Ghiani, G., Improta, G.: An efficient transformation of the generalized vehicle routing problem. Eur. J. Oper. Res. **122**, 11–17 (2000)
17. Golden, B.: Recent developments in vehicle routing. In: White, W.W. (ed.) Computers and Mathematical Programming. Proceedings of the Bicentennial Conference on Mathematical Programming, pp. 233–240 (1978)
18. Hà, M.H., Bostel, N., Langevin, A., Rousseau, L.M.: An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. Comput. Oper. Res. **43**, 9–19 (2014)
19. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods. Springer, Berlin (1993)
20. ILOG, Inc. ILOG CPLEX: High-performance software for mathematical programming and optimization (2012)
21. Ioachim, I., Desrosiers, J., Soumis, F., Bélanger, N.: Fleet assignment and routing with schedule synchronization constraints. Eur. J. Oper. Res. **119**, 75–90 (1999)
22. Lübbecke, M.E., Desrosiers, J.: Selected topics in column generation. Oper. Res. **53**, 1007–1023 (2005)
23. Marsten, R.E., Hogan, W.W., Blackenship, J.W.: The boxstep method for large-scale optimization. Oper. Res. **23**, 389–405 (1975)
24. Moccia, L., Cordeau, J.-F., Laporte, G.: An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. J. Oper. Res. Soc. **63**, 232–244 (2012)
25. Quttineh, N.-H., Larsson, T.: Military aircraft mission planning: efficient model-based metaheuristic approaches. Optim. Lett. (forthcoming)
26. Quttineh, N.-H., Larsson, T., Van den Bergh, J., Beliën, J.: A Time-Indexed Generalized Vehicle Routing Model for Military Aircraft Mission Planning. Operations Research Proceedings (2014) (forthcoming)
27. Quttineh, N.-H., Larsson, T., Lundberg, K., Holmberg, K.: Military aircraft mission planning: a generalized vehicle routing model with synchronization and precedence. EURO J. Transp. Logist. **2**, 109–127 (2013)
28. Redjem, R., Kharraja, S., Xie, X., Marcon, E.: Routing and scheduling of caregivers in home health care with synchronized visits. In: 9th International Conference of Modeling, Optimization and Simulation – MOSIM'12 (2012)
29. Royset, J.O., Carlyle, W.M., Wood, R.K.: Routing military aircraft with a constrained shortest-path algorithm. Mil. Oper. Res. **3**, 31–52 (2009)
30. Schumacher, C., Chandler, P.R., Pachter, M., Pachter, L.S.: Optimization of air vehicles operations using mixed-integer linear programming. J. Oper. Res. Soc. **58**, 516–527 (2007)
31. Sigurd, M., Pisinger, D., Sig, M.: Scheduling transportation of live animals to avoid the spread of diseases. Transp. Sci. **38**, 197–209 (2004)
32. Westerlund, A., Göthe-Lundgren, M., Larsson, T.: A stabilized column generation scheme for the traveling salesman subtour problem. Discret. Appl. Math. **154**, 2212–2238 (2006)
33. Wilhelm, W.E.: A technical review of column generation in integer programming. Optim. Eng. **2**, 159–200 (1995)
34. Zabarankin, M., Uryasev, S., Murphey, R.: Aircraft routing under the risk of detection. Nav. Res. Logist. **53**, 728–747 (2006)

On Deterministic Diagonal Methods for Solving Global Optimization Problems with Lipschitz Gradients

Yaroslav D. Sergeyev and Dmitri E. Kvasov

Abstract In this chapter, a global optimization problem is considered where both the objective function $f(x)$ and its gradient $\nabla f(x)$ are multidimensional black-box functions. It is supposed that $\nabla f(x)$ satisfies the Lipschitz condition over the search hyperinterval with an unknown Lipschitz constant K. Different techniques for estimating K are presented and their advantages and disadvantages are emphasized. In what regards exploring the multidimensional search domain, several adaptive partitioning strategies are discussed that can be applied in Lipschitz global optimization methods: (1) one-point-based algorithms evaluating the objective function and its gradient at one point within each subregion; (2) diagonal partitions where $f(x)$ and $\nabla f(x)$ are evaluated at two points within each subregion; (3) more complex partitions based, for instance, on simplices or auxiliary functions of various nature. This chapter deals with diagonal deterministic methods that show a promising performance both in tests and applications. Several geometric methods based on diagonal partitions and auxiliary functions are presented and compared on eight hundred of differentiable problems randomly produced by the GKLS-generator of classes of test functions.

Keywords Black-box optimization • Lipschitz global optimization methods • Diagonal partitions - Lipschitz gradients

1 Problem Statement

The global optimization problem with a differentiable objective function having the Lipschitz gradient (with an unknown Lipschitz constant) is an important

Y.D. Sergeyev • D.E. Kvasov (✉)
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria, Via P. Bucci, Cubo 42C – 87036 Rende (CS), Italy

Software Department, Lobachevsky State University, Nizhni Novgorod, Russia
e-mail: yaro@si.dimes.unical.it; kvadim@si.dimes.unical.it

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_16

class of Lipschitz global optimization problems (see, e.g., the references given in [5, 44, 48]). This class of problems can be defined as follows:

$$f^* = f(x^*) = \min_{x \in D} f(x), \tag{1}$$

$$\|\nabla f(x') - \nabla f(x'')\| \leq K\|x' - x''\|, \quad x', x'' \in D, \quad 0 < K < \infty, \tag{2}$$

where

$$D = [a, b] = \{x \in R^n : a(j) \leq x(j) \leq b(j), 1 \leq j \leq n\}. \tag{3}$$

It is assumed here that the objective function $f(x)$ can be black-box and multiextremal, its gradient $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x(1)}, \frac{\partial f(x)}{\partial x(2)}, \ldots, \frac{\partial f(x)}{\partial x(n)}\right)^T$ (which can be itself an expensive black-box vector-function) can be calculated during the search, and $\nabla f(x)$ is Lipschitz-continuous with some unknown constant K, $0 < K < \infty$, over D.

Problem (1)–(3) is frequently met in engineering applications (see, e.g., [10, 11, 23, 33, 43, 48]), for instance, in electrical engineering design (see, e.g., [46, 48]). Each evaluation of both $f(x)$ and $\nabla f(x)$ at a point $x \in D$ (this operation is often called 'trial') is supposed to be a time-consuming operation, therefore, it is desirable to obtain a solution to the stated problem by evaluating $f(x), \nabla f(x)$ at the less possible number of trial points.

Different methods for solving problem (1)–(3) have been proposed (see, e.g., [2–6, 21, 30, 43–45, 47, 48]). They can be distinguished either by the strategy of exploration of the search hyperinterval D from (3) or by the mode in which information about the Lipschitz constant K from (2) is obtained.

In exploring the multidimensional search domain, various adaptive partitioning strategies can be applied. For example, one-point-based algorithms subsequently subdivide the search region in smaller ones and evaluate the objective function and its gradient at one point within each subregion (see, e.g., [4, 6, 21]). Diagonal partitions that evaluate $f(x)$ and $\nabla f(x)$ at two points within each subregion are very interesting for practical applications with expensive black-box functions (see, e.g., [23, 33, 42, 43]). More complex partitions, based, for instance, on simplices or auxiliary functions of various nature can be also used (see, e.g., [5, 12, 27, 30, 31, 47, 49]).

In specifying the Lipschitz constant K from (2), several ways can be considered: this constant can be given a priori (see, e.g., [1, 2, 4]); its adaptive estimates (local or global) can be obtained during the search (see, e.g., [6, 12, 22, 26, 36, 37, 43, 45, 48]); multiple estimates of the Lipschitz constant can be also used (see, e.g., [20, 21]).

This chapter discusses some promising deterministic methods constructed in the framework of the diagonal approach for solving problem (1)–(3). An introduction into the diagonal technique is given in Sect. 2 where different strategies for partitioning the search domain D from (3) are considered. Several methods belonging to this framework are presented in Sect. 3. Results of their numerical comparison on eight hundred of multidimensional GKLS test functions (see [7]) with respect to different criteria are given in the conclusive Sect. 4.

2 Diagonal Partition Strategies

In global optimization, a variety of techniques for an iterative partition of the admissible hyperinterval D from (3) into a set of $M(l)$ hyperintervals D_i, $1 \leq i \leq M(l)$ (where $l \geq 1$ is the iteration counter) can be used during the search for the global minimum (see, e.g., [15, 17, 33, 40, 43]). Over each hyperinterval D_i, $1 \leq i \leq M(l)$, the approximation of $f(x)$ is based on results obtained from evaluating $f(x)$ (and $\nabla f(x)$) at some points $x \in D$. For example, the widely used global optimization method DIRECT [17] involves partitioning with evaluation of $f(x)$ at the central points of hyperintervals.

In this chapter, the main attention is devoted to diagonal algorithms introduced in [32, 33] for derivative-free global optimization problems with the Lipschitz objective functions. This approach has attractive theoretical properties and has proved to be efficient in solving applied problems. In these algorithms, both the objective function $f(x)$ and its gradient $\nabla f(x)$ (diagonal methods using gradients have been proposed, e.g., in [8, 21, 39, 43, 45]) are evaluated—independently of the problem dimension—only at the vertices corresponding to the main diagonal $[a_i, b_i]$ of each generated hyperinterval D_i (either at both the vertices, as usually done, see [8, 18, 24, 33, 40, 43, 45], or at only one of them, as used in [21, 41, 43]). Results of these trials are then used to estimate the function behavior over the generated hyperintervals and to select a hyperinterval (or a set of hyperintervals) for the further subdivision.

Particularly, at every iteration $l \geq 1$ of a diagonal method the 'merit' of each hyperinterval D_i, $1 \leq i \leq M(l)$, of the current partition is estimated. A higher 'merit' of hyperinterval D_i corresponds to a higher possibility that the global minimizer x^* of $f(x)$ from (1) belongs to D_i. The 'merit' is measured by a real-valued function R_i called characteristic (introduced in its general form within the framework of divide-the-best algorithms [38, 43] which diagonal methods belong to). In order to calculate the characteristic R_i of a multidimensional hyperinterval D_i, some one-dimensional characteristics can be used as prototypes (see, e.g., [9, 13, 14]). They can be applied to the one-dimensional segment being the main diagonal $[a_i, b_i]$ of the hyperinterval D_i. A hyperinterval having the 'best' characteristic (e.g., the smallest lower bound of $f(x)$ over the hyperintervals) is partitioned (by hyperplanes passing through some chosen point on the main diagonal) by means of a diagonal partition strategy and new trials are performed, thus improving the approximation of the solution to problem (1)–(3).

For example, in Fig. 1, both $f(x)$ and $\nabla f(x)$ are evaluated at points a_i and b_i of a hyperinterval D_i. The function $f(x)$ is approximated along the diagonal $[a_i, b_i]$ of D_i by means of a specially constructed auxiliary function $\varphi_i(\gamma)$, $\gamma \in [a_i, b_i]$. The minimum value φ_i^* (at the point γ_i^*) of this approximating function estimates the lower bound of $f(x)$ over the segment $[a_i, b_i]$. Then, the estimate φ_i^* is multiplied by a coefficient in order to be a lower estimate for $f(x)$ not only over the diagonal but also over the whole multidimensional hyperinterval D_i. This modified lower estimate is accepted as the characteristic R_i of the hyperinterval D_i.

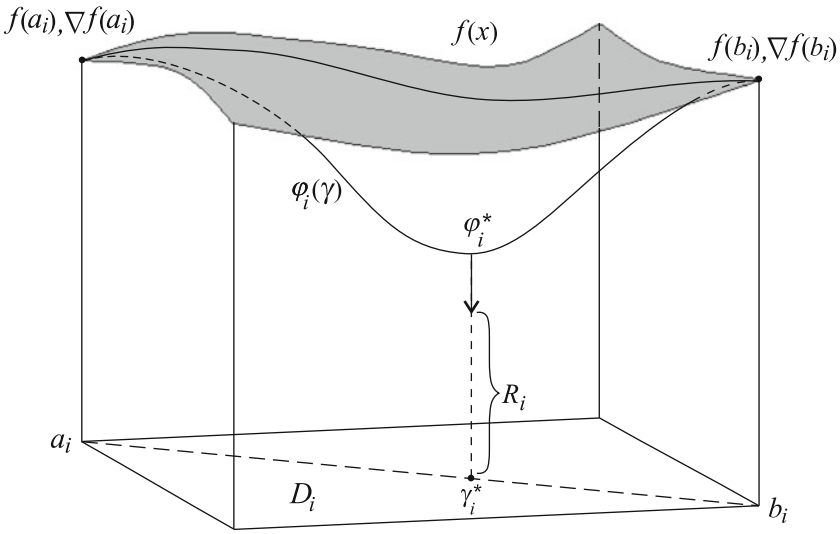


Fig. 1 Obtaining characteristic R_i of a hyperinterval D_i in a diagonal global optimization method

As already mentioned, the diagonal algorithms belong to the class of divide-the-best methods [38]. Therefore, on the one hand, general convergence theory developed for analysis of divide-the-best algorithms (described, e.g., in [38, 43]) can be successfully applied to the analysis of the diagonal algorithms too. On the other hand, the diagonal approach provides a natural generalization (see, e.g., [33, 40, 43, 47]) of many one-dimensional algorithms to the multidimensional case.

The concrete choice of both the partition strategy and the function R_i determines a particular diagonal method for solving problem (1)–(3). Some considerations regarding diagonal partition strategies are given in this section, while the issues related to the characteristics of diagonal methods are presented in the next section.

Two strategies for hyperinterval partitioning are often used in diagonal algorithms to partition a hyperinterval $D_t = D_{t(l)}$ at iteration l (see Fig. 2): 2^n-*Partition* (see, e.g., [16, 17, 24, 29, 33]) and *Bisection* (see, e.g., [8, 16, 24, 33, 39]). In 2^n-Partition strategy (see Fig. 2a), the chosen (at the iteration l) hyperinterval D_t is partitioned into 2^n new hyperintervals [n is the problem dimension from (3)] generated by the intersection of the boundary of D_t and the hyperplanes that contain a point γ_t^* belonging to the main diagonal of D_t and are parallel to the boundary hypersurfaces of D_t. In Bisection strategy (see Fig. 2b), the hyperinterval D_t is subdivided into two hyperintervals (not necessarily of the same volume) by a hyperplane passing through γ_t^* (without performing a trial in this point) and orthogonal to the longest edge of D_t. In both the cases, the new trials are performed at both vertices of the main diagonal of each newly generated hyperintervals

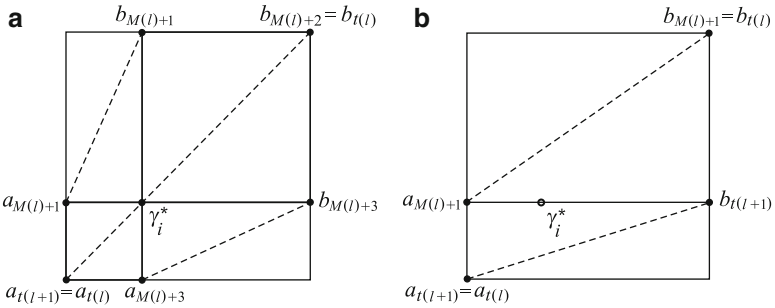


Fig. 2 Partition of a two-dimensional hyperinterval $D_t = [a_t, b_t]$ executed at iteration l of a diagonal method based on 2^n-Partition (**a**) and Bisection (**b**) strategies (*black dots* indicate trial points)

(that is, $2 \times 2^n - 3$ and two new trials are performed—except those already executed at the vertices $a_{t(l)}$ and $b_{t(l)}$—after subdividing a hyperinterval by 2^n-Partition and Bisection strategies, respectively, as indicated by black dots in Fig. 2).

The usage of the diagonal approach has the goal to decrease the computational efforts needed to describe the behavior of $f(x)$ over every hyperinterval D_i by evaluating $f(x), \nabla f(x)$ at only two vertices of D_i instead of evaluating it at all 2^n vertices. Therefore, the evaluation of both $f(x)$ and $\nabla f(x)$ at $2^{n+1} - 3$ points during each iteration can impose too high computational demand on solving problem (1)–(3) by a diagonal algorithm using 2^n-Partition strategy (see Figs. 2a and 3a). Therefore, the diagonal Bisection partition strategy seems to be more computationally economic for solving expensive global optimization problems (as confirmed, e.g., by [8, 24, 39]).

However, as shown, e.g., in [19, 40, 43], Bisection strategy also generates too many trial points in the course of the algorithm execution (irrespective of the form of the characteristic that determines which hyperinterval is to be subdivided at each iteration). This fact is mainly due to the following reason (see [40] for more details). When the number of hyperintervals in the current partition of D increases, each hyperinterval contains more and more trial points on its edges. These points can be closely spaced or even coincide but it is not easy to establish efficiently this adjacency. Thus, both $f(x)$ and $\nabla f(x)$ are often re-evaluated at the same or close points during different iterations while it would be sufficient to evaluate them only at one of these points (2^n-Partition strategy has similar drawbacks).

In Fig. 3, an illustration of this redundancy problem for both the 2^n-Partition and Bisection strategies is presented. Each digit is the number of the iteration at which $f(x)$ and $\nabla f(x)$ have been evaluated at the corresponding point (at the first iteration, the trials have been performed at the vertices a and b). It can be seen that several trial points generated at different iterations are very close to one another (these points are circled in Fig. 3). Notice also the two coincident points corresponding to the seventh and eighth iterations in Fig. 3b: a diagonal method using the Bisection

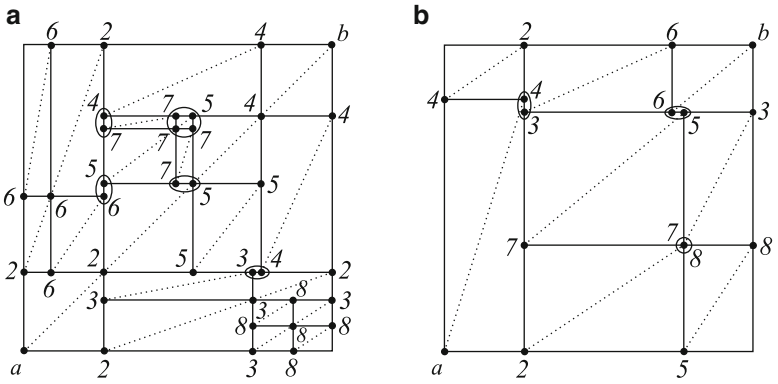


Fig. 3 Partition of a two-dimensional hyperinterval D after eight iterations executed by a diagonal algorithm based on 2^n-Partition (**a**) and Bisection (**b**) strategies (*black dots* indicate trial points, redundant points are *circled*)

strategy would evaluate $f(x)$ and $\nabla f(x)$ at the same point twice and store the results of the trials twice in different areas of the computer memory.

This redundancy slows down a diagonal algorithm using traditional partition strategies because of the high computational resources needed for the functions evaluations. Therefore, traditional diagonal schemes often do not fulfill the requirements of computational efficiency in black-box global optimization.

The so-called non-redundant diagonal partition strategy, proposed in [40], allows one to avoid the computational redundancy of traditional diagonal partition strategies. It trisects a hyperinterval by two parallel hyperplanes into three hyperintervals of equal volume, performing new trials exactly at two new points. This strategy produces regular meshes of trial points in such a way that one vertex where $f(x)$ and $\nabla f(x)$ are evaluated can belong to several hyperintervals (up to 2^n). As shown, e.g., in [18, 40, 43], a special indexation of the hyperintervals can be proposed in order to efficiently establish links between hyperintervals having common facets but generated during different iterations. In this way, the time-consuming procedure of the functions evaluation is replaced by a significantly faster operation of reading (up to 2^n times) the functions values obtained at some previous iterations and saved in a special database. Hence, the non-redundant partition strategy considerably speeds up the search and also leads to saving computer memory. It is particularly important that the advantages of this strategy become more pronounced when the problem dimension n increases.

Let us give an example of the application of a diagonal algorithm based on the non-redundant partition strategy. In Fig. 4, partitions of a two-dimensional admissible region D produced by such an algorithm at several initial iterations $l \geq 1$ are presented starting from the first two trials at the points a and b. As in Fig. 3, black dots represent trial points and the numbers around these dots indicate iterations at which these trial points have been generated. Hyperintervals shown in light gray

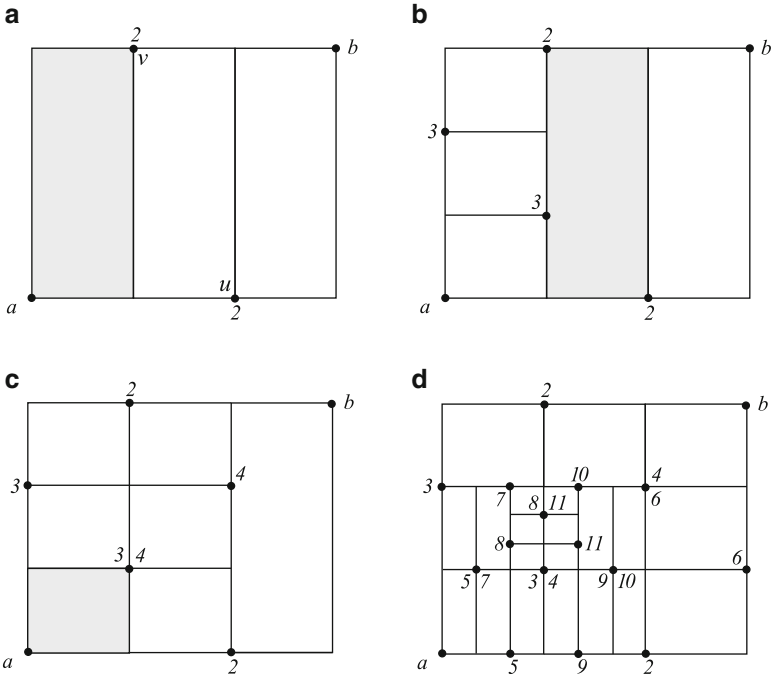


Fig. 4 An example of subdivisions by the non-redundant diagonal partition strategy (see the explanation in the text)

are chosen for the subdivision at the current iteration of the algorithm. In Fig. 4a, the situation after the first two iterations is presented. Particularly, at the second iteration, the hyperinterval D is partitioned into three new hyperintervals of equal volume. This subdivision is performed by two hyperplanes orthogonal to the longest edge of D that pass through points u and v (see Fig. 4a). Thus, at the third iteration, three smaller hyperintervals are generated (see Fig. 4b). As one can see from Fig. 4c, the trial point of the fourth iteration coincides with the point 3 at which the trial has already been executed. Therefore, there is no need to perform a new (costly) evaluation of $f(x)$ and $\nabla f(x)$ at this point, since the values obtained at the previous iteration can be used. These values can be stored in a specially designed vertex database and is simply retrieved on demand without re-evaluations of the functions. For example, Fig. 4d illustrates the situation after 11 iterations. It can be seen from this figure that 21 hyperintervals have been generated by 17 trial points and four times the functions values have been retrieved from the database. The more higher is the problem dimension, the more pronounced is the computational advantage of the considered diagonal partition strategy.

Surprisingly (see [41]), this strategy developed for the diagonal methods can be also applied for the one-point-based algorithms (consequently, such algorithms will be called diagonal too). Instead of evaluating $f(x)$ and $\nabla f(x)$ at two vertices

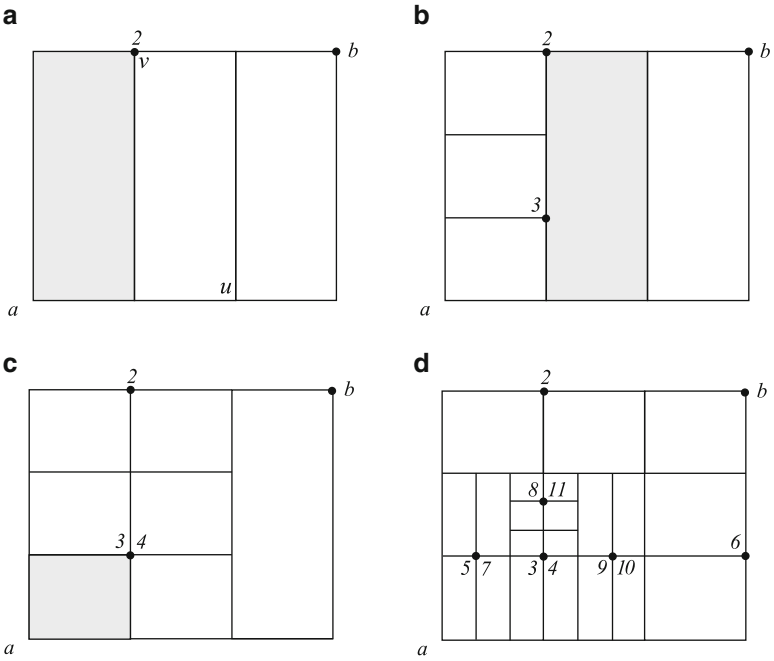


Fig. 5 One-point partition strategy based on the non-redundant diagonal strategy (see the explanation in the text)

u and v (as in Fig. 4a), it is possible to do this initially for one of the vertices (let us take the vertex b) of the region D and then at the corresponding vertex v (see Fig. 5a) during every splitting (the point u is used just for partitioning goals). The operation of verifying whether a trial has been already performed at a vertex is done by using the same efficient procedure as for the non-redundant diagonal strategy (compare Figs. 4a–d and 5a–d). For example, in Fig. 5d the distribution of trial points generated during the first 11 iterations by an algorithm using this one-point-based strategy is given: here, 21 hyperintervals have been produced by only seven trial points. It can be seen from this figure that each hyperinterval contains exactly one vertex where $f(x)$ and $\nabla f(x)$ have been evaluated.

In the next section, some methods for solving problem (1)–(3) based on the mentioned partition strategies will be briefly described.

3 Diagonal Global Optimization Methods

As known, the Lipschitz condition (2) can be used to obtain the lower bound of the global minimum value (1) of the objective function $f(x)$ at each iteration $l \geq 1$ of a Lipschitz global optimization algorithm, thus allowing one to construct

numerical methods and to prove their convergence in a unified manner in the framework of the divide-the-best scheme. The methods of this type form the class of geometric algorithms that are based on constructing, updating, and improving auxiliary functions (see function $\varphi_i(\gamma)$ in Fig. 1) built by using an estimate of the Lipschitz constant K from (2) (see, e.g., [1, 2, 20–22, 26, 45, 47]). Similar ideas are used in many other surrogate-based optimization methods (see, e.g., [5]).

Since at each point $x \in D$ from (3) it is possible to evaluate both the objective function and its gradient, more information about the problem is available (especially, regarding its local properties expressed by the gradient values). The usage of this information allows one to construct auxiliary functions that fit closely the objective function and to accelerate the global search.

A particular attention in this contribution is given to three diagonal algorithms proposed in the literature for solving problem (1)–(3). They differ both in the used partition strategies (see Sect. 2) and estimates of the Lipschitz constant for the gradient of $f(x)$. In the context of these diagonal methods, the directional derivative $f'(c_i)$ evaluated at a vertex c_i along the main diagonal $[a_i, b_i]$ of D_i (i. e., either $c_i = a_i$ or $c_i = b_i$) is used in what follows instead of the gradient vector $\nabla f(c_i)$:

$$f'(c_i) = \left(\sum_{j=1}^{n} \frac{\partial f(c_i)}{\partial x(j)} (b_i(j) - a_i(j)) \right) / \Delta_i, \tag{4}$$

where

$$\Delta_i = \|a_i - b_i\| = \sqrt{\sum_{j=1}^{n} (a_i(j) - b_i(j))^2} \tag{5}$$

is the length of the main diagonal of hyperinterval $D_i \subset D$, n is from 3.

The first algorithm (denoted hereafter as BISECTION) is from [8]; it is based on Bisection diagonal partition and at each its iteration l uses estimate of local Lipschitz constants K_i over hyperintervals D_i, $1 \leq i \leq M(l)$. The second method (denoted hereafter as SMOOTHD) is from [45]; the non-redundant diagonal partition strategy is applied in this algorithm together with the usage of smooth auxiliary functions to estimate $f(x)$ over main diagonals of hyperintervals. Finally, the third method (denoted hereafter as MULTK) is from [21]; it uses the non-redundant one-point-based diagonal strategy and multiple estimates of the Lipschitz constant K. The latter algorithm is also characterized by a smart usage of the local information during its work, whilst the first two algorithms are taken in their basic versions that can be further improved (for example, by adopting the local tuning technique from [24, 34, 35, 37] in the SMOOTHD scheme, by involving both the methods in a two-phase approach as in [21, 31, 42], and so on).

More details on these methods can be found in [8, 21, 45], respectively. Here, let us only give an insight into the construction of diagonal auxiliary functions employed in the methods to assign the hyperintervals characteristics.

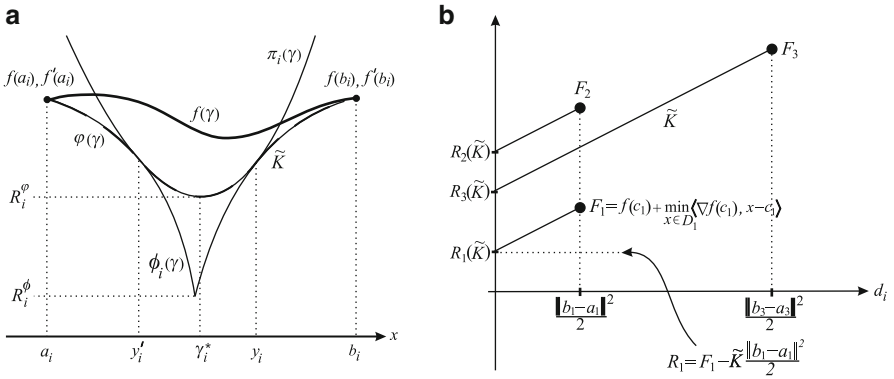


Fig. 6 Obtaining characteristics R_i for hyperintervals $D_i = [a_i, b_i]$ in the methods BISECTION and SMOOTHD (**a**) and MULTK (**b**) (see formulae (9), (10), and (12), respectively)

Given an estimate $\tilde{K} \geq K$ of the Lipschitz constant K from (2) and taking into account the Taylor expansion of $f(x)$ limited to the second order term, the next inequality can be obtained for $f(x)$ in $D_i = [a_i, b_i] \subset D$, once a trial at vertex $c_i \in [a_i, b_i]$ is executed:

$$f(x) \geq f(c_i) + \langle \nabla f(c_i), (x - c_i) \rangle - 0.5\tilde{K}\|x - c_i\|^2, \quad x \in D_i, \tag{6}$$

where $\langle \cdot, \cdot \rangle$ is the scalar product, $\| \cdot \|$ is the Euclidean norm in R^n.

Inequality (6) can be then used in the framework of geometric diagonal algorithms to construct auxiliary functions along main diagonals of hyperintervals (see Fig. 1) and to calculate the hyperintervals characteristics (see Fig. 6).

For example, in the BISECTION method, a non-smooth piecewise quadratic auxiliary function $\phi_i(\gamma)$, $\gamma \in [a_i, b_i]$ (see Fig. 6a) can be considered along the main diagonal $[a_i, b_i]$ of D_i [as consequence of (6)], with estimates $\tilde{K}_i$ of local Lipschitz constants K_i in (6) over D_i, $1 \leq i \leq M(l)$, calculated as follows:

$$\tilde{K}_i = 0.5\tilde{K}(1 + (\hat{K}_i/\tilde{K})^2), \tag{7}$$

where $\tilde{K}$ is an estimate of the Lipschitz constant K from (2) over the whole search domain D:

$$\tilde{K} = \begin{cases} r \max\limits_{1 \leq i \leq M(l)} \hat{K}_i, & \text{if } \max\limits_{1 \leq i \leq M(l)} \hat{K}_i > 0, \\ 1, & \text{otherwise}, \end{cases} \tag{8}$$

$r > 1$ is the reliability parameter of the method, and

$$\hat{K}_i = \max \begin{cases} |f'(a_i) - f'(b_i)|/\Delta_i, \\ 2[-(f(a_i) - f(b_i)) + f'(b_i)\Delta_i]/\Delta_i^2, \\ 2[(f(a_i) - f(b_i)) - f'(a_i)\Delta_i]/\Delta_i^2, \end{cases} \quad 1 \leq i \leq M(l),$$

with $f'(a_i), f'(b_i)$ and Δ_i calculated by (4) and (5), respectively.

Then, the characteristic R_i of hyperinterval $D_i = [a_i, b_i]$, $1 \le i \le M(l)$, corresponds to the minimum of the function $\phi_i(\gamma)$ for $\gamma \in [a_i, b_i]$ (see value R_i^{ϕ} in Fig. 6a):

$$R_i = f(b_i) + f'(b_i)\delta_i - 0.5\tilde{K}_i\delta_i^2, \tag{9}$$

where

$$\delta_i = \frac{-(f(a_i) - f(b_i)) + f'(a_i)\Delta_i + 0.5\tilde{K}_i\Delta_i^2}{\tilde{K}_i\Delta_i + (f'(a_i) - f'(b_i))}.$$

The BISECTION method iteratively subdivides a hyperinterval (and, hence, adaptively performs new trials) with the smallest characteristic by using Bisection partition strategy until its volume (or the length of its main diagonal) becomes smaller than the accuracy related to a preset constant $\varepsilon > 0$.

Let us now consider the SMOOTHD method from [45] which extends the one-dimensional techniques from [37] to the multidimensional case through the diagonal approach. Particularly, in [37], it has been demonstrated (for one-dimensional problems) how to obtain smooth auxiliary functions making them closer to $f(x)$ than the previously used ones (as, for example, non-smooth functions of the BISECTION method) and, therefore, accelerating the global search (see also [1, 28] where similar constructions are discussed). A general scheme describing one-dimensional methods using smooth bounding procedures has been presented in [37] with several approaches for the Lipschitz constant estimation (including a global estimate $\tilde{K}$ of the Lipschitz constant K considered here).

As observed in [37, 45], the objective function is above the function $\phi_i(\gamma)$ for all $\gamma \in (y_i', y_i)$ (see Fig. 6a) because due to (2) its curvature along this segment is bounded by a parabola

$$\pi_i(\gamma) = 0.5\tilde{K}\gamma^2 + B_i\gamma + C_i.$$

The unknowns y_i', y_i, B_i, and C_i can be determined by solving the following system of equations:

$$\begin{cases} \phi_i(y_i') = \pi_i(y_i'), \\ \phi_i(y_i) = \pi_i(y_i), \\ \phi_i{}'(y_i') = \pi_i{}'(y_i'), \\ \phi_i{}'(y_i) = \pi_i{}'(y_i). \end{cases}$$

Here, the first equation provides the coincidence of $\phi_i(\gamma)$ and $\pi_i(\gamma)$ at the point y_i' and the third one provides the coincidence of their derivatives $\phi_i{}'(\gamma)$ and $\pi_i{}'(\gamma)$ at the same point. The second and fourth equations provide the fulfilment of these conditions at the point y_i.

Thus, once the values y'_i, y_i, B_i, and C_i are determined (see [45] for details), it may be concluded that the following function

$$\varphi_i(\gamma) = \begin{cases} \phi_i(\gamma), \ \gamma \in [a_i, y'_i) \cup [y_i, b_i], \\ \pi_i(\gamma), \ \gamma \in [y'_i, y_i), \end{cases}$$

is a smooth piecewise quadratic auxiliary function over the main diagonal $[a_i, b_i]$ of D_i. Its minimum value

$$\varphi_i(\gamma_i^*) = f(b_i) - f'(b_i)\Delta_i - 0.5\tilde{K}\Delta_i^2 + \tilde{K}y_i^2 - 0.5\tilde{K}(\gamma_i^*)^2 = R_i^{\varphi} \tag{10}$$

obtained at the point

$$\gamma_i^* = 2y_i - \tilde{K}^{-1}f'(b_i) - \Delta_i$$

corresponds to the characteristic R_i of hyperinterval D_i, $1 \le i \le M(l)$ (as usual, $f'(a_i), f'(b_i)$ and Δ_i are calculated by (4) and (5), respectively).

The current estimate $\tilde{K}$ of the Lipschitz constant K from (2) for the objective function gradient is found in the SMOOTHD method as:

$$\tilde{K} = \begin{cases} r \max\limits_{1 \le i \le M(l)} \hat{K}_i, \text{ if } \max\limits_{1 \le i \le M(l)} \hat{K}_i > \xi = 10^{-6}, \\ r\,\xi, \qquad\qquad \text{otherwise}, \end{cases} \tag{11}$$

where $r > 1$ is the reliability parameter of the method, ξ is a small positive value (it ensures the correct algorithm execution when the values $\hat{K}_i$ are too small), and $\hat{K}_i$ are calculated as

$$\hat{K}_i = \frac{|2(f(a_i) - f(b_i)) + (f'(a_i) + f'(b_i))\Delta_i| + d_i}{\Delta_i^2},$$

with

$$d_i = \{[2(f(a_i) - f(b_i)) + (f'(a_i) + f'(b_i))\Delta_i]^2 + (f'(b_i) - f'(a_i))^2\Delta_i^2\}^{\frac{1}{2}}.$$

The local tuning technique [leading, e.g., to a formula similar to (7)] can be also used for estimating the Lipschitz constant K.

The stopping criterion of the SMOOTHD method is similar to that of the BISECTION method, i. e., the method stops when the volume (or the length of the main diagonal) of a hyperinterval with the smallest characteristic until becomes smaller than the accuracy related to a preset constant $\varepsilon > 0$.

Finally, let us consider the MULTK method from [21]. With respect to the BISECTION and SMOOTHD methods, that do not use any local improvement technique, the MULTK method consists of the two explicitly defined phases: (1) an exploration phase, at which an examination of large hyperintervals (possibly

located far away from the current best point) is performed in order to capture new subregions with better function values; (2) a record improvement phase, at which the algorithm tries to better inspect the subregion around the record point. The record improvement phase reflects the already well-established fact in global optimization affirming the benefits of the local improvement during the global search (see, e.g., the references given in [15, 20, 25, 26, 43, 47]).

The MULTK method generalizes the approach proposed in [20] for the one-dimensional problems by means of the one-point-based diagonal scheme and uses during its work a series of non-smooth (discontinuous) piecewise quadratic auxiliary functions. Each of these functions [based on the right-hand part of inequality (6)] corresponds to a particular estimate $\tilde{K}$ of the Lipschitz constant K taken from zero to infinity. Given the estimate $\tilde{K}$, a lower bound R_i of the objective function values over hyperinterval $D_i = [a_i, b_i]$ (i. e., the hyperinterval characteristic) can be calculated as

$$R_i = F_i - \tilde{K} d_i, \tag{12}$$

where

$$F_i = f(c_i) + \min_{x \in D_i} \langle \nabla f(c_i), (x - c_i) \rangle, \qquad d_i = 0.5 \|b_i - a_i\|^2. \tag{13}$$

In (13), the value of c_i is equal either to a_i or to b_i, depending on the partition scheme used (a_i-point-based scheme has been used in [21] while b_i-point-based scheme is considered in this contribution). The minimum in (13) is attained at one of the vertices of D_i and is easily found (see [21]).

Both a hyperinterval D_i and the respective characteristic R_i from (12) can be represented in a two-dimensional diagram similar to that proposed in [17, 42] for derivative-free methods. In this diagram (see Fig. 6b), the dot with the coordinates (F_i, d_i) from (13) corresponds to the hyperinterval D_i, $1 \le i \le M(l)$. The characteristic R_i of the hyperinterval D_i can be graphically obtained as the vertical coordinate of the intersection point of the line passed through the point D_i with the slope $\tilde{K}$ and the vertical coordinate axis (see Fig. 6 where a partition of D into three hyperintervals D_1, D_2, and D_3 is represented). In this way, the hyperinterval with the best characteristic for a given estimate of the Lipschitz constant is easily identified. For example, in Fig. 6b, given the estimate $\tilde{K}$, hyperinterval D_1 has the smallest characteristic with respect to the other hyperintervals. If a higher estimate $\hat{K} > \tilde{K}$ of the Lipschitz constant K is considered, the characteristic of hyperinterval D_1 still remains better than that of hyperinterval D_2, but it becomes worse than the characteristic of hyperinterval D_3 with respect to $\hat{K}$, because $R_3(\hat{K}) < R_1(\hat{K})$ for a $\hat{K}$ sufficiently higher than $\tilde{K}$ (see Fig. 6b).

Since the exact Lipschitz constant K for the gradient of $f(x)$ (or its valid overestimate) is unknown in the stated global optimization problem, a set of possible estimates for K from zero to infinity is used in the MULTK method, thus introducing a set of nondominated hyperintervals (i.e., hyperintervals with the smallest characteristics (12) for some particular estimate of the Lipschitz constant

for the gradient of $f(x)$). For example, in Fig. 6b the hyperintervals D_1 and D_3 are nondominated. The set of nondominated hyperintervals corresponds graphically to the lower-right convex hull of the set of dots representing the hyperintervals of the current partition of D and can be efficiently found by applying algorithms for identifying the convex hull of the dots (see, e.g., [17, 43]). A number of nondominated hyperintervals are then subdivided at the exploration phase of the MULTK method. The MULTK method stops when a preset trials budget is depleted.

To conclude, let us report results of convergence analysis of the three methods.

Theorem 1 (BISECTION [8], SMOOTHD [45]). *For any function $f(x)$ with the gradient satisfying the Lipschitz condition (2) with the constant K, $0 < K < \infty$, and for the algorithms* BISECTION *and* SMOOTHD*, there exists a value r^* of the reliability parameter r from (8) and (11) such that for any $r \geq r^*$ the infinite ($\varepsilon = 0$ in the methods stopping criterion) sequence of trial points, generated by these methods during minimization of $f(x)$, will converge only to the global minimizers of $f(x)$.*

Theorem 2 (MULTK [21]). *For any point $x \in D$ and any $\rho > 0$ there exist an iteration number $l(\rho) \geq 1$ and a trial point $x' = c_{i(l)}$, $l > l(\rho)$, generated by the* MULTK *method with the unlimited trial budget, such that $\|x - x'\| < \rho$.*

Since all the three methods belong to the class of divide-the-best algorithms (see [38]), results of Theorems 1 and 2 can be obtained as particular cases of the general convergence study of divide-the-best algorithms from [38].

Both the BISECTION and SMOOTHD methods have strong convergence properties. The reliability of these methods is improved by increasing the reliability parameter r [see formulae (8) and (11)]. If this parameter is correctly defined (either after experimental or theoretical investigations of problem (1)–(3), see [45]), the methods converge to the points of the global minimum of $f(x)$. As this parameter decreases, the search rate increases, but the probability of convergence to a local point within some hyperinterval other than the global minimizer of $f(x)$ grows as well. In fact, if a value of r smaller than r^* is used in these algorithms, they can converge (see the general analysis executed for divide-the-best methods in [38]) to a local minimizer of $f(x)$ or to the boundary of a subregion of D corresponding to the best characteristics. This situation indicates the necessity to increase the value of the reliability parameter, i. e., it is also a practical hint for the choice of r.

In contrast, the MULTK method manifests the so-called everywhere dense convergence and has no any internal stopping criterion, useful to demonstrate the goodness of the found solution to the problem (important in many applied problems).

4 Results of Numerical Comparison

In this section, we present numerical results performed to compare the described algorithms between themselves and with the DIRECT algorithm from [17]. The latter one has been taken as a benchmark method due to its extensive use in

solving applied global optimization problems (see, e.g., [5]). It uses the center-sampling partitioning strategy and works with a set of Lipschitz constants for the objective function $f(x)$ from (1), without evaluating $\nabla f(x)$. The implementation of this method from http://www4.ncsu.edu/~ctk/SOFTWARE/DIRECTv204.tar.gz was used in the experiments.

In our numerical experiments, eight GKLS D-type classes of dimensions $n = 2$, 3, 4, and 5 were considered, each of 100 continuously differentiable functions, produced by the GKLS-generator (see [7]) as in [45]. For each particular problem dimension n a "simple" and a "hard" classes were taken for the comparison to highlight the influence of the problems complexity on the methods behavior (see [42, 45] for a detailed description of the classes).

As in [21, 31, 42, 45], the global minimizer $x^* \in D$ from (1) was considered to be found when a method generated a trial point x' inside a hyperinterval with a vertex x^* and the volume smaller than the volume of the initial hyperinterval $D = [a,b]$ multiplied by an accuracy coefficient ε, $0 < \varepsilon \leq 1$, i. e.,

$$|x'(j) - x^*(j)| \leq \sqrt[n]{\varepsilon}(b(j) - a(j)), \quad 1 \leq j \leq n, \tag{14}$$

where n is from (3). This coefficient was taken equal to 10^{-4} for $n = 2$, 10^{-6} for $n = 3$ and $n = 4$, and 10^{-7} for $n = 5$. The tested algorithm was stopped either when the maximal number of trials $P_{\max}$ equal to 100,000 was reached, or when condition (14) was satisfied (see [42] and the previous section for a discussion about different stopping criteria in global optimization methods).

The balancing parameter equal to 10^{-4} was used in the DIRECT method, as recommended by many authors (see [5, 17, 31]). Several values of the reliability parameter r were used in the BISECTION method, starting from the initial value $r = 1.1$ (the maximal values of this parameter equal to 1.7, 3.3, 2.1, 2.8, 3.2 were chosen for the first five classes and were increased up to 10.0 for the last three GKLS classes used in the experiments). An adaptive reliability parameter

$$r = \bar{r} + C/l$$

was used in the SMOOTHD method, as investigated in [45], where $l \geq 1$ is the iteration counter and C is a positive constant (set in relation to the problems dimension as 50 for $n = 2$, 100 for $n = 3$, 150 for $n = 4$, and 200 for $n = 5$). The maximal values of the coefficient $\bar{r}$ were equal to 2.8, 5.8, 3.6, 4.3, 5.8, 6.6, 4.1, and 7.8 for the GKLS functions from the first class to the last one, respectively. More details on the choice of the reliability parameter can be found, e.g., in [45].

The methods were compared on the following criteria (see [42, 43] for more details):

Criterion 1. (a) maximal number of trial points required for a method to satisfy condition (14) for all 100 functions of a particular test class and (b) the corresponding number of generated hyperintervals (see Tables 1 and 2).

Table 1 Maximal number of trial points for GKLS test functions (Criterion 1a)

n	ε	Class	DIRECT	BISECTION	SMOOTHD	MULTK
2	10^{-4}	Simple	1,159	1,106	332	257
2	10^{-4}	Hard	3,201	2,084	893	323
3	10^{-6}	Simple	12,507	6,963	3,092	2,091
3	10^{-6}	Hard	$\gg$100,000 (4)	13,230	4,807	3,875
4	10^{-6}	Simple	$\gg$100,000 (4)	82,435	20,059	18,054
4	10^{-6}	Hard	$\gg$100,000 (11)	$>$100,000 (1)	50,699	23,769
5	10^{-7}	Simple	$>$100,000 (2)	$>$100,000 (1)	10,912	17,543
5	10^{-7}	Hard	$\gg$100,000 (42)	$>$100,000 (7)	93,245	85,047

Table 2 Corresponding number of hyperintervals for GKLS test functions (Criterion 1b)

n	ε	Class	DIRECT	BISECTION	SMOOTHD	MULTK
2	10^{-4}	Simple	1,159	553	599	861
2	10^{-4}	Hard	3,201	1,042	1,669	1,119
3	10^{-6}	Simple	12,507	3,482	10,389	11,813
3	10^{-6}	Hard	$\gg$100,000	6,615	16,749	23,653
4	10^{-6}	Simple	$\gg$100,000	41,218	119,345	187,717
4	10^{-6}	Hard	$\gg$100,000	$>$50,000	188,219	230,797
5	10^{-7}	Simple	$>$100,000	$>$50,000	89,343	257,583
5	10^{-7}	Hard	$\gg$100,000	$>$50,000	321,913	1,496,629

Table 3 Average number of trial points for GKLS test functions (Criterion 2a)

n	ε	Class	DIRECT	BISECTION	SMOOTHD	MULTK
2	10^{-4}	Simple	198.89	432.75	151.11	74.75
2	10^{-4}	Hard	1,063.78	707.03	404.79	162.11
3	10^{-6}	Simple	1,117.70	3,369.76	1,011.00	783.49
3	10^{-6}	Hard	$\gg$6,322.65	4,934.85	1,756.18	618.32
4	10^{-6}	Simple	$\gg$11,282.89	4,061.15	4,598.97	3,512.92
4	10^{-6}	Hard	$\gg$29,540.12	$>$59,581.96	7,276.23	6,127.09
5	10^{-7}	Simple	$>$6,956.97	$>$40,772.45	4,281.42	3,583.20
5	10^{-7}	Hard	$\gg$72,221.24	$>$50,223.86	33,246.18	19,688.68

Criterion 2. (a) average number of trial points required for a method to satisfy condition (14) for 100 functions of a particular test class and (b) the corresponding number of generated hyperintervals (see Tables 3 and 4).

Results based on the first criterion are mainly influenced by minimization of the most difficult functions of a class whilst the second criterion deals with average data of the class. The number of generated hyperintervals provides an important characteristic of any partition algorithm for solving problem (1)–(3) and corresponds to the qualitative examination of the search domain D during the work of the method.

Table 4 Corresponding number of hyperintervals for GKLS test functions (Criterion 2b)

n	ε	Class	DIRECT	BISECTION	SMOOTHD	MULTK
2	10^{-4}	Simple	198.89	216.38	253.64	233.02
2	10^{-4}	Hard	1,063.78	353.52	736.20	540.94
3	10^{-6}	Simple	1,117.70	1,684.88	3,051.16	2,890.72
3	10^{-6}	Hard	$\gg$6,322.65	2,467.43	5,628.76	4,388.24
4	10^{-6}	Simple	$\gg$11,282.89	20,305.58	22,913.25	33,160.98
4	10^{-6}	Hard	$\gg$29,540.12	$>$29,790.98	49,083.08	61,810.38
5	10^{-7}	Simple	$>$6,956.97	$>$20,386.23	32,588.80	44,163.18
5	10^{-7}	Hard	$\gg$72,221.24	$>$123,920.70	96,764.18	272,344.60

Results of the numerical comparison of the methods with respect to the used criteria with the eight GKLS test classes are shown in Tables 1, 2, 3, and 4. The notation "$>$100,000 (j)" in Tables 1 and 3 means that after 100,000 function trials the method under consideration was not able to solve j problems (the DIRECT method was not able to solve many of these problems even after 1,000,000 trials as reported in [42], with the consequent increase of the average values in Tables 3 and 4—this fact is marked by '$\gg$; it should be, however, noticed that each trial in the DIRECT method is computationally lighter than that of the other three methods). The data from unsolved problems were not taken in computation of the averages in Tables 3 and 4.

As demonstrated by the results of the extensive numerical experiments performed, the usage of the gradient information together with the efficient partitioning strategy allows one to obtain a serious acceleration in comparison with the derivative-free DIRECT method on the given classes of test problems.

According to Tables 1, 2, 3, and 4, the algorithms SMOOTHD and MULTK based on the non-redundant diagonal partition strategy require much fewer trials than the BISECTION method to ensure a thorough examination of the search domain for the considered test classes. The advantage of these methods becomes more pronounced as the problem dimension grows or the problem complexity increases. This confirms the observations of Sect. 2 regarding the redundancy of Bisection partition strategy, traditionally used in diagonal algorithms.

In its turn, the MULTK method behaves generally better than the SMOOTHD algorithm on the used GKLS classes. This result is explained by the usage of a strong local improvement phase incorporated in the MULTK method, particularly suitable for the stopping criterion (14) (it should be mentioned that this algorithm has no an internal stopping criterion as in the SMOOTHD method). Adding such a record improvement phase to the SMOOTHD method (tested here in its basic version as given in [45]) would significantly speed up the algorithm execution in terms of the function trials, as suggested by investigations performed in [21, 26, 47].

Acknowledgements This work was partially supported by the INdAM–GNCS 2014 Research Project of the Italian National Group for Scientific Computation of the National Institute for Advanced Mathematics "F. Severi."

References

1. Baritompa, W., Cutler, A.: Accelerations for global optimization covering methods using second derivatives. J. Glob. Optim. **4**(3), 329–341 (1994)
2. Breiman, L., Cutler, A.: A deterministic algorithm for global optimization. Math. Program. **58**(1–3), 179–199 (1993)
3. Cartis, C., Fowkes, J.M., Gould, N.I.M.: Branching and bounding improvements for global optimization algorithms with Lipschitz continuity properties. J. Glob. Optim. **61**(3), 429–457 (2015)
4. Evtushenko, Y.G., Posypkin, M.A.: A deterministic approach to global box-constrained optimization. Optim. Lett. **7**(4), 819–829 (2013)
5. Floudas, C.A., Pardalos, P.M. (eds.): Encyclopedia of Optimization (6 Volumes), 2nd edn. Springer, Berlin (2009)
6. Fowkes, J.M., Gould, N.I.M., Farmer, C.L.: A branch and bound algorithm for the global optimization of Hessian Lipschitz continuous functions. J. Glob. Optim. **56**(4), 1791–1815 (2013)
7. Gaviano, M., Lera, D., Kvasov, D.E., Sergeyev, Y.D.: Algorithm 829: software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans. Math. Softw. **29**(4), 469–480 (2003)
8. Gergel, V.P.: A global optimization algorithm for multivariate function with Lipschitzian first derivatives. J. Glob. Optim. **10**(3), 257–281 (1997)
9. Gergel, V.P., Sergeyev, Y.D.: Sequential and parallel algorithms for global minimizing functions with Lipschitzian derivatives. Comput. Math. Appl. **37**(4–5), 163–179 (1999)
10. Gillard, J.W., Zhigljavsky, A.A.: Optimization challenges in the structured low rank approximation problem. J. Glob. Optim. **57**(3), 733–751 (2013)
11. Gillard, J.W., Zhigljavsky, A.A.: Stochastic algorithms for solving structured low-rank matrix approximation problems. Commun. Nonlinear Sci. Numer. Simul. **21**(1–3), 70–88 (2015)
12. Gorodetsky, S.Y.: Paraboloid triangulation methods in solving multiextremal optimization problems with constraints for a class of functions with Lipschitz directional derivatives. Vestnik of Lobachevsky State University of Nizhni Novgorod **1**(1), 144–155 (2012) (in Russian)
13. Grishagin, V.A.: Operating characteristics of some global search algorithms. In: Problems of Stochastic Search, vol. 7, pp. 198–206. Zinatne, Riga (1978) (in Russian)
14. Grishagin, V.A., Sergeyev, Y.D., Strongin, R.G.: Parallel characteristic algorithms for solving problems of global optimization. J. Glob. Optim. **10**(2), 185–206 (1997)
15. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization, vol. 1. Kluwer Academic Publishers, Dordrecht (1995)
16. Horst, R., Tuy, H.: Global Optimization – Deterministic Approaches. Springer, Berlin (1996)
17. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**(1), 157–181 (1993)
18. Kvasov, D.E.: Multidimensional Lipschitz global optimization based on efficient diagonal partitions. 4OR Q. J. Oper. Res. **6**(4), 403–406 (2008)
19. Kvasov, D.E., Sergeyev, Y.D.: Multidimensional global optimization algorithm based on adaptive diagonal curves. Comput. Math. Math. Phys. **43**(1), 42–59 (2003)
20. Kvasov, D.E., Sergeyev, Y.D.: A univariate global search working with a set of Lipschitz constants for the first derivative. Optim. Lett. **3**(2), 303–318 (2009)
21. Kvasov, D.E., Sergeyev, Y.D.: Lipschitz gradients for global optimization in a one-point-based partitioning scheme. J. Comput. Appl. Math. **236**(16), 4042–4054 (2012)

22. Kvasov, D.E., Sergeyev, Y.D.: Univariate geometric Lipschitz global optimization algorithms. Numer. Algebra Contr. Optim. **2**(1), 69–90 (2012)
23. Kvasov, D.E., Sergeyev, Y.D.: Deterministic approaches for solving practical black-box global optimization problems. Adv. Eng. Softw. **80**, 58–66 (2015)
24. Kvasov, D.E., Pizzuti, C., Sergeyev, Y.D.: Local tuning and partition strategies for diagonal GO methods. Numer. Math. **94**(1), 93–106 (2003)
25. Lera, D., Sergeyev, Y.D.: An information global minimization algorithm using the local improvement technique. J. Glob. Optim. **48**(1), 99–112 (2010)
26. Lera, D., Sergeyev, Y.D.: Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. SIAM J. Optim. **23**(1), 508–529 (2013)
27. Lera, D., Sergeyev, Y.D.: Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants. Commun. Nonlinear Sci. Numer. Simul. **23**(1–3), 328–342 (2015)
28. MacLagan, D., Sturge, T., Baritompa, W.: Equivalent methods for global optimization. In: Floudas, C.A., Pardalos, P.M. (eds.) State of the Art in Global Optimization, pp. 201–212. Kluwer Academic Publishers, Dordrecht (1996)
29. Molinaro, A., Pizzuti, C., Sergeyev, Y.D.: Acceleration tools for diagonal information global optimization algorithms. Comput. Optim. Appl. **18**(1), 5–26 (2001)
30. Paulavičius, R., Žilinskas, J.: Simplicial Global Optimization. Springer, New York (2014)
31. Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J.: Globally-biased DISIMPL algorithm for expensive global optimization. J. Glob. Optim. **59**(2–3), 545–567 (2014)
32. Pintér, J.D.: Extended univariate algorithms for *N*-dimensional global optimization. Computing **36**(1–2), 91–103 (1986)
33. Pintér, J.D.: Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications). Kluwer Academic Publishers, Dordrecht (1996)
34. Sergeyev, Y.D.: An information global optimization algorithm with local tuning. SIAM J. Optim. **5**(4), 858–870 (1995)
35. Sergeyev, Y.D.: A one-dimensional deterministic global minimization algorithm. Comput. Math. Math. Phys. **35**(5), 705–717 (1995)
36. Sergeyev, Y.D.: A method using local tuning for minimizing functions with Lipschitz derivatives. In: Bomze, I.M., Csendes, T., Horst, R., Pardalos, P.M. (eds.) Developments in Global Optimization, pp. 199–216. Kluwer Academic Publishers, Dordrecht (1997)
37. Sergeyev, Y.D.: Global one-dimensional optimization using smooth auxiliary functions. Math. Program. **81**(1), 127–146 (1998)
38. Sergeyev, Y.D.: On convergence of "Divide the Best" global optimization algorithms. Optimization **44**(3), 303–325 (1998)
39. Sergeyev, Y.D.: Multidimensional global optimization using the first derivatives. Comput. Math. Math. Phys. **39**(5), 711–720 (1999)
40. Sergeyev, Y.D.: An efficient strategy for adaptive partition of *N*-dimensional intervals in the framework of diagonal algorithms. J. Optim. Theory Appl. **107**(1), 145–168 (2000)
41. Sergeyev, Y.D.: Efficient partition of *N*-dimensional intervals in the framework of one-point-based algorithms. J. Optim. Theory Appl. **124**(2), 503–510 (2005)
42. Sergeyev, Y.D., Kvasov, D.E.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. SIAM J. Optim. **16**(3), 910–937 (2006)
43. Sergeyev, Y.D., Kvasov, D.E.: Diagonal Global Optimization Methods. FizMatLit, Moscow (2008) (in Russian)
44. Sergeyev, Y.D., Kvasov, D.E.: Lipschitz global optimization. In: Cochran, J.J. et al. (eds.) Wiley Encyclopedia of Operations Research and Management Science, vol. 4, pp. 2812–2828. Wiley, New York (2011)
45. Sergeyev, Y.D., Kvasov, D.E.: A deterministic global optimization using smooth diagonal auxiliary functions. Commun. Nonlinear Sci. Numer. Simul. **21**(1–3), 99–111 (2015)

46. Sergeyev, Y.D., Daponte, P., Grimaldi, D., Molinaro, A.: Two methods for solving optimization problems arising in electronic measurements and electrical engineering. SIAM J. Optim. **10**(1), 1–21 (1999)
47. Sergeyev, Y.D., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. Springer, New York (2013)
48. Strongin, R.G., Sergeyev, Y.D.: Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms. Kluwer Academic Publishers, Dordrecht (2000)
49. Zhigljavsky, A.A., Žilinskas, A.: Stochastic Global Optimization. Springer, New York (2008)

Optimization of Design Parameters for Active Control of Smart Piezoelectric Structures

Georgios Stavroulakis, Georgia Foutsitzi, and Christos Gogos

Abstract The objective of this work is to design an optimal controller for plate structures to control their response under the influence of external excitation. The finite element method based on the Mindlin–Reissner plate theory has been extended to incorporate the piezoelectric effects. A genetic algorithm is applied to find the optimal placement of piezoelectric actuators and input voltages for static shape control. The objective function is the error in transverse displacements between the desired and the achieved shape.

In addition, the optimal placement of actuators and sensors for vibration control of laminated plates is studied. The objective taken into consideration is the controllability index, which is the singular value decomposition of a control matrix as can be found at the bibliography. The index measures the input energy required to achieve the desired structural control using piezoelectric actuators.

Finally, the linear quadratic regulator (LQR) closed loop control is applied to study the control effectiveness. A comparison is made between the optimal locations of piezoactuators obtained through controllability index and a nonoptimal case.

Keywords Optimal actuator/ sensor placement • Mindlin-Reissner plate • Finite element method • Genetic algorithm • Linear quadratic regulator control

1 Introduction

Smart composite structures are receiving increasing attention due to their significant potential applicability in various industrial and research areas. Excellent sensing and actuating capabilities of piezoelectric materials made them the most practical, smart

G. Stavroulakis (✉)
Department of Production Engineering and Management, Technical University of Crete, Institute of Computational Mechanics and Optimization University Campus, 73100 Chania, Greece
e-mail: gestavr@dpem.tuc.gr

G. Foutsitzi • C. Gogos
Technological Educational Institution of Epirus, TEI Campus, Psathaki, 48100 Preveza, Greece
e-mail: gfoutsi@teiep.gr; cgogos@teiep.gr

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_17

materials to integrate with laminated composite structures. Optimal distribution of the piezoelectric material in the structure to induce controlled actuation has been a subject of interest in recent years.

In static shape control applications, the objective is to optimize control parameters such as the placement of actuators and the applied electric voltage so that the desired shapes are achieved or best matched. Optimization of such parameters and configurations of piezoelectric actuators for acquiring efficient and precise shape control have been an interesting subject of research in recent years [1, 2, 7]. The review article by Irschik [9] describes relevant applications of static and dynamic shape control of structures by piezoelectric actuation.

In addition, piezoelectric material technology has found application in active vibration control of smart structures. As it is well known in the area of smart structures and control engineering, the performance of active vibration control depends not only upon the control law but also on the placement of piezoelectric sensors and actuators. Optimal actuator placement is the topic of a large portion of the recent work in smart structures' optimization and is based on several criteria (from rigorous measures of controllability and observability, coming from optimal control theory, to more intuitive measures of deviation from a desired response) [8, 10, 12]. A "technical review" until 2010 that presents the "most used" optimization criteria by researchers for optimal placement of piezoelectric sensors and actuators on a smart structure can be found in [6].

Following the theory of optimal control is only possible for the design of linear systems with linear control feedback. The more general approach for the design of controllers that cover nonlinear mechanical models and nonlinear control laws is based on numerical optimization. Active control applications of smart composite structures usually involve large, nonconvex, integer programming problems that are discrete in nature. Global optimization algorithms (such as genetic algorithms, evolutionary algorithms, and particle swarm optimization algorithms) are often suitable for these types of problems. Moreover, they are simple to implement when compared to other optimization techniques, allowing for their application in a wide range of problems in this area of study [2, 4, 7, 8, 10, 12]. GA methods are computationally effective in finding the global optimal solution for a nonconvex function which has no derivative. Several authors have already used this method to optimize the actuator and sensor locations, for example [4, 10, 12].

The objective of this work is to design an optimal controller to control the response of plate structures to control its response under the influence of external excitation. The finite element method based on the Mindlin–Reissner plate theory has been extended to incorporate the piezoelectric layers that are used as sensors and actuators. An improved genetic algorithm is applied to find the optimal placement of piezoelectric actuators and input voltages for static shape control. The objective function is the error in transverse displacements between the desired shape and the achieved one.

In addition, the optimal placement of actuators and sensors for vibration control of laminated plates is studied. The objective taken into consideration is the

controllability index, which is the singular value decomposition of a control matrix [13]. The index measures the input energy required to achieve the desired structural control using piezoelectric actuators.

Finally, the linear quadratic regulator (LQR) closed loop control is applied to study the control effectiveness. A comparison is made between the optimal locations of piezo-actuators obtained through controllability index.

2 Finite Element Modeling of Piezoelectric Smart Structures

Consider a flexible structure with *Na* piezoelectric actuators and *Ns* piezoelectric sensors bonded to its upper surface and lower surface, respectively. From finite element analysis, the equations of motion and charge equilibrium of the system can be written as follows:

$$\begin{aligned} [M]\left\{\ddot{d}\right\} + [K_{uu}]\{d\} + \left[K_{u\phi}\right]\{\phi\} &= \{F_m\} \\ \left[K_{\phi u}\right]\{d\} + \left[K_{\phi\phi}\right]\{\phi\} &= \left\{F_q\right\} \end{aligned} \tag{1}$$

where, $\{d\}$ and $\{\phi\}$ are the global mechanical and electrical DoF vectors, $[M]$ is the global mass matrix, $[K_{uu}]$, $\left[K_{u\phi}\right] = \left[K_{u\phi}\right]^T$, and $\left[K_{\phi\phi}\right]$ are the global mechanical stiffness, mechanical-electrical coupling stiffness, and dielectric stiffness matrices respectively. $\{F_m\}$ and $\left\{F_q\right\}$ are the respective global mechanical and electrical loads vectors.

Next, we assume that the electrical DoF vector in Eq. (1) can be divided into the actuating and sensing DoFs, $\{\phi\}_e = \{\phi_a, \phi_s\}^T$, where the subscripts a and s denote the actuating and sensing capabilities. Hence, considering open-circuit electrodes, and in that case $\left\{F_q\right\} = 0$, the nonspecified potential differences in (1) can be statically condensed and the equations of motion and charge equilibrium become

$$\begin{aligned} [M]\left\{\ddot{d}\right\} + [K_{uu}]\{d\} &= \{F_m\} - \left[K_{u\phi}\right]_a\{\phi\}_a \\ \{\phi\}_s &= -\left[K_{\phi\phi}\right]_s^{-1}\left[K_{\phi u}\right]_s\{d\} \end{aligned} \tag{2}$$

where $[K^*_{uu}] = [K_{uu}] - \left[K_{u\phi}\right]_s\left[K_{\phi\phi}\right]_s^{-1}\left[K_{\phi u}\right]_s$.

Equation (2) can be used in smart structure applications such as vibration control and static or dynamic shape control. In shape control applications, the piezoelectric layers are used as actuators. In addition the time-dependent momentum forces become zero. Thus, all the electrical degrees are considered as known quantities and the coupled Eq. (2) reduce to pure mechanical ones:

$$\left[K_{u\phi}\right]\{d\} = \{F_m\} - \{F_{el}\} \tag{3}$$

where $\{F_{el}\} = \left[K_{u\phi}\right]\{\phi\}$ is the electrical force vector due to the actuation.

2.1 State Space Formulation of the Modal Control Problem

The application of the active control methods in dynamic structural problem requires the use of a state space model. Before we obtain this kind of equations, a mode superposition method is adopted to obtain an approximate reduced-order dynamic model of the system with uncoupled equations of motion in the modal coordinates. This step is essential for two reasons. First modal analysis gives a lot of qualitative information about the dynamical response of a system and helps us design an effective smart system. In addition, reduction of the size is essential for the design of the controller. Hence $\{d(t)\}$ can be approximated by

$$\{d\} \approx \sum_{i=1}^{r} \Phi_j \eta_j = [\Phi]\{\eta\} \tag{4}$$

where $[\Phi] = [\Phi_1, \Phi_2, \cdots, \Phi_r]$ is the truncated modal matrix and $[\eta] = [\eta_1, \eta_2, \cdots, \eta_r]$ is the modal coordinate vector. Substituting Eqs. (4) into (2) leads to

$$\{\ddot{\eta}\} + \left[\Omega^2\right]\{\eta\} = [\Phi]^T\{F_m\} - [\Phi]^T\left[K_{u\phi}\right]_\alpha\{\phi\}_\alpha \tag{5}$$

$$\{\phi\}_s = -\left[K_{\phi\phi}\right]_s^{-1}\left[K_{u\phi}\right]_s[\Phi]^T\{\eta\} \tag{6}$$

Also using the modal approach, structural damping can be easily introduced as

$$\{\ddot{\eta}\} + [\Lambda]\{\dot{\eta}\} + [\Omega]\{\eta\} = [\Phi]^T\{F_m\} - [\Phi]^T\left[K_{u\phi}\right]_\alpha\{\phi\}_\alpha \tag{7}$$

where $[\Lambda]$ is a diagonal modal damping matrix with the generic term $2\xi_i\omega_i$, where ξ_i is the modal damping ratio and ω_i the undamped natural frequency of the i_{th} mode. For control design, Eqs. (5) and (6) are transformed into state-space forms as follows:

$$\begin{aligned} \{\dot{x}\} &= [A]\{x\} + [B]\left\{u_\phi\right\} + \{f\} \\ \{\phi\}_s &= \{y\} = \{C\}\{x\} \end{aligned} \tag{8}$$

where $\{x\} = \{\eta, \dot{\eta}\}^T$ is the state vector, $[A]$ is the system matrix, $[B]$ is the control matrix, $\{f\}$ is the disturbance input vector, and $\left\{u_\phi\right\} = \{\phi\}_\alpha$ is the control input to the actuator. These matrices are given by

$$[A] = \begin{bmatrix} [0] & [I] \\ \left[-\Omega^2\right] & [\Lambda] \end{bmatrix} \qquad [B] = \begin{bmatrix} [0] \\ -[\Phi]^T\left[K_{u\phi}\right]_a \end{bmatrix}$$

$$\{f\} = \begin{bmatrix} [0] \\ [\Phi]^T\{F_m\} \end{bmatrix} \qquad [C] = \left[-\left[K_{\phi\phi}\right]_s^{-1}\left[K_{\phi u}\right]_s[\Phi]\,[0]\right] \tag{9}$$

2.2 *Controllability Index for Actuator Location*

In this section, the controllability index is introduced which is based on the state equation (8). This index will be used to determine the optimal placements of piezoelectric actuators in vibration control of plate structures.

From the expressions of matrices $[A]$ and $[B]$ in state equation (8), it is clear that all control designs depend on the placement and size of the piezoelectric actuators as well as the vibration modes used in the modal analysis.

Performing the singular value decomposition of control matrix $[B]$ we get the singular values

$$S = \begin{bmatrix} \sigma_1 & \dots & 0 \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ & & \sigma_{n_p} \\ 0 & \dots & 0 \end{bmatrix}, n_p < n \tag{10}$$

where n_p is the number of piezoelectric actuators and n is the number of modes used in the modal analysis. The magnitude of σ_i is a function of the location and size of piezoelectric actuators. Wang and Wang [13] proposed that the controllability index (CI) is defined by

$$\hat{\Omega} = \prod_{i=1}^{n_p} \sigma_i \tag{11}$$

The higher the CI, the lower the power consumption required for control, i.e., the better the control effectiveness. The index measures the input energy required to achieve a desired structural control by the piezoelectric actuators.

3 Optimal Controller Design

In the following, controller parameters such as actuator locations and/or actuation voltages are optimized in shape control and vibration control of plate structures.

The shape control of smart plate structures has been formulated as an optimization problem to find out the optimal actuator locations as well as the actuator voltages in a plate structure.

In addition the vibration control of the plate has been formulated as an optimization problem with design variables the locations of collocated piezoelectric actuators–sensors (S/As) pairs.

In this work, it is assumed that every actuator covers exactly the length of one element and possible actuator locations are described through a vector a of binary variables whose elements a_i are 1 to indicate the presence and 0 the absence of an actuator.

3.1 Optimization of Actuator Location and Voltages in Shape Control

Shape control consists of finding a set of design variables (i.e., actuator size, location, and voltages) that minimizes the difference between achieved and desired shape under certain constraints.

When considering plate elements, the shape of a structure is mainly described by the shape of its midplane, which itself is described by the transverse displacement of the finite element mesh nodes. Therefore, the error between the predefined displacement field function and the achieved displacement field can be defined as the sum of the errors at the r nodal points, and the fitness or objective function, f, is then given by

$$f = \sum_{i=1}^{r} |w_i - w_i^d| \tag{12}$$

where w_i^d is the desired nodal displacement value and r is the number of concerned displacements.

In general the transverse displacement is a function of the electric potential, the location, and the size of actuators. In this framework, the shape control problem studied consists of finding the optimal position of actuators and the applied voltages for a given number of actuators, which minimize the cost function f under the following constraint

$$\begin{aligned} \phi_{min} \leq \phi_i \leq \phi_{max} \\ a_i \in \{0,1\} \\ \sum_{i=1}^{n_{ac}} a_i \leq p \end{aligned} \tag{13}$$

where ϕ_i is the actuation voltage of the ith actuator and ϕ_{min} and ϕ_{max} are the lower and upper saturation voltages of the actuators, n_{ac} is the number of actuators, p is a given number which is lower than the number of finite elements and a_i takes the value 1 or 0 to indicate the presence or the absence of an actuator.

The Mixed Integer Problem that arises is highly nonlinear and is solved using a genetic algorithm procedure in order to accommodate two different types of information: the location of each piezoelectric element and the voltage needed to apply to each of them.

3.2 Optimization of Actuator Location in Vibration Control

In active control of smart structures, the placement of actuators and sensors on the structure is a very important issue in order to achieve the most effective actuation. Optimal placement of sensors and actuators over a structure might vary for different criteria. Next, in order to propose performance criteria for S/A locations, the maximization of CI defined above has been considered. In the current analysis, CI has been used as a measure of control effectiveness. The problem is to determine the optimal placements a_i of piezoelectric collocated sensor–actuator pairs on the plate which maximize the CI:

$$f_1 = \hat{\Omega} \tag{14}$$

3.3 Optimization Implementation Using Genetic Algorithms

The two optimization problem formulated in Sects. 3.2 and 3.3 can be stated in the following general form:

Find a design variable $x = (\phi, a)$ for the first case or find a design variable $x = (a)$ for the second case that

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{Subject To } x_i^{min} \le x_i \le x_i^{max},\ i = 1, 2, \cdots, N_d \\ &&& g_j(x) \le 0, \quad j = 1, 2, \cdots, N \end{aligned} \tag{15}$$

where $f(x)$ is the objective function, x is the vector of design variables x_j, $g_j(x)$ are the N inequality behavioral constraint equations, x_i^{min} and x_i^{max} are the lower and upper limits of the design variables, respectively, and N_d is the total number of design variables.

Both problems are solved using genetic algorithms (GAs). Genetic Algorithms are a general purpose global optimization method that are known for their wide applicability to several engineering optimization problems [5]. For the case of the shape control the GA is fed with a number of initial solutions generated by the local search optimization method Great Deluge [3]. For the case of the vibration control problem GA achieved equally good results starting from a random initial population.

The computational difficulty of the shape control problem of plate structures is discussed below. The objective is to simultaneously determine locations and voltages for a set of actuators so that the difference between the desired and the achieved shapes is minimized. The underlying equations of the problem formulation are non-linear. Furthermore, the search domain space is quite large since k out of n binary variables have to be designed assuming value 1 while the rest ones

should assume value 0. In this study $n = 36$ and $k = 10$, which accounts for over 250 million possible combinations. Additionally, k continuous variables have to be decided corresponding to the voltage applied to each actuator. Each of these k values is bounded by problem-specific lower and upper limits which are for this work 0 and 100 V, respectively. So, the domain space of the problem is substantially big and combined with the fact that the evaluation of the plate position for each possible solution involves a nontrivial amount of computations results in an interesting optimization problem.

For the case of the vibration control of plate structures the problem is simpler since no voltages have to be determined but only the actuator locations. One might expect that the best solution might exhibit symmetry since from a symmetrical initial shape of the plate we try to return to another symmetrical target shape. If this hypothesis is correct that means that by activating some actuators on the left half of the plate and their mirrors at the right half of it, this will then result to the best possible solution. Since the problem in this case is rather small it is possible to try a full enumeration of all possible combinations for the left half of the plate, mirror the solution to the right half and evaluate each solution. In our experiments where $n = 36$ and $k = 10$, only 8,568 cases had to be examined. Nevertheless, the solution that resulted by picking the best among the above combinations was not as good as the solution that the GA provided.

4 Numerical Applications

In this section, we present two applications about active control of rectangular plates with piezoelectric patches. The MatLab software package combined with the Global Optimization Toolbox was used for the development of the algorithm. The algorithm is able to solve the optimization problems stated in the previous paragraphs given the number of actuators/sensors alongside with the plate dimensions and properties. The computer code developed makes no assumption of linearity between the displacements and the electric voltages; thus, it can be used for non-linear models as well.

After validating the present formulation with the existing results in the literature, the problem of optimal designing the variables for active control of the cantilever plate shown in Fig. 1 is considered. The plate has a dimension of $200 \times 200 \times 1.2$ mm. The composite plate consists of four composite substrate layers and two outer PZT layers. The stacking sequence of the substrate is antisymmetric angle-ply $[-45^\circ/45^\circ/-45^\circ/45^\circ]$. The substrate is made of T300/976 graphite–epoxy composite and the PZT is PZT G1195N. The elastic moduli and Poisson's ratios for the graphite–epoxy material and the piezoceramic are those proposed in [11]: $E_1 = 150.0\,\text{GPa}$, $E_2 = E_3 = 9.0\,\text{GPa}$, $G12 = 7.1\,\text{GPa}$, $G2_3 = G_{13} = 2.5\,\text{GPa}$, $v_{12} = v_{13} = v_{23} = 0.3$ and $E_1^p = 63.0\,\text{GPa}$, $G_{12}^p = G_{23}^p = G_{13}^p = 24.2\,\text{GPa}$, $v_{12}^p = v_{23}^p = v_{13}^p = 0.3$. The piezoelectric constants are $d_{13} = d_{23} = 254 \times 10^{-12}$ and

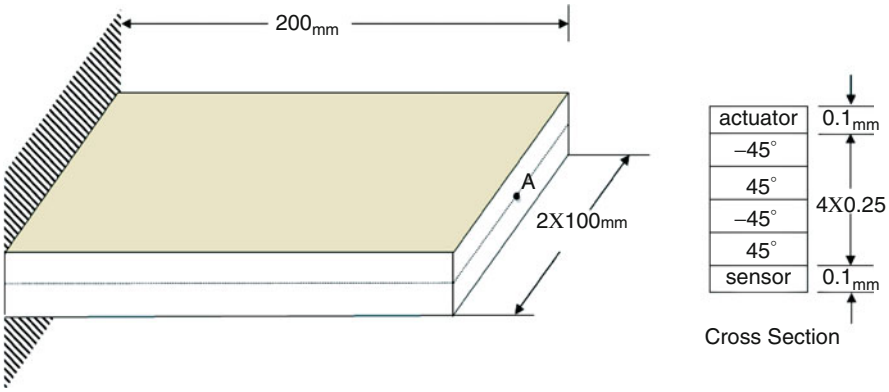


Fig. 1 The cantilever plate

$d_{42} = d_{51} = 584 \times 10^{-12}$. The total thickness of the composite plate is 1 mm and each layer has the same thickness (0.25 mm). The thickness of each PZT is $hp = 0.1$ mm. The length of one piezoelectric patch is assumed to be equal to the length of one finite element. For the finite element analysis the beam is divided into 36 (6×6) elements and the total number of piezoelectric patches is assumed to be ten. Next, all the 36 elements are candidates for locations of the ten piezoelectric actuator patches of length equal to finite element mesh. The stiffness and mass of the piezoelectric patches are taken into account in the model.

In all applications, plates have the same size and the same piezoelectric material is used.

4.1 Shape Control

In this section the optimal voltages and locations of ten actuators are calculated for shape control of a plate with two different kinds of disturbances. In the case of shape control, all piezoceramics on the upper and lower surfaces of the plate are used as actuators. The lower limit of the voltage is set to be 0 V and the upper limit is set to be 100 V (limit imposed due to depoling of actuators). The desired shape is given by $w^d(x) = 0$.

4.1.1 A Cantilever Plate Subject to a Point Force at the Tip

The problem of the cantilever laminated composite plate when a point load of $1N$ is applied at point A is studied in this section. The genetic algorithms were run using the following parameters: Generations = 1,000, Population = 100, EliteCount = 4. Table 1 shows the optimal solutions for placement of the actuators and the corresponding optimal values of the actuated voltages. We observe that the

Table 1 Optimal locations of ten actuators on the cantilever plate for the desired shape $w^d(x) = 0$

Optimal locations	Corresponding optimal voltages
2, 7, 8, 9, 10, 19, 21, 25, 27, 35	100, 100, 100, 100, 100, 100, 100, 85.74, 100, 100

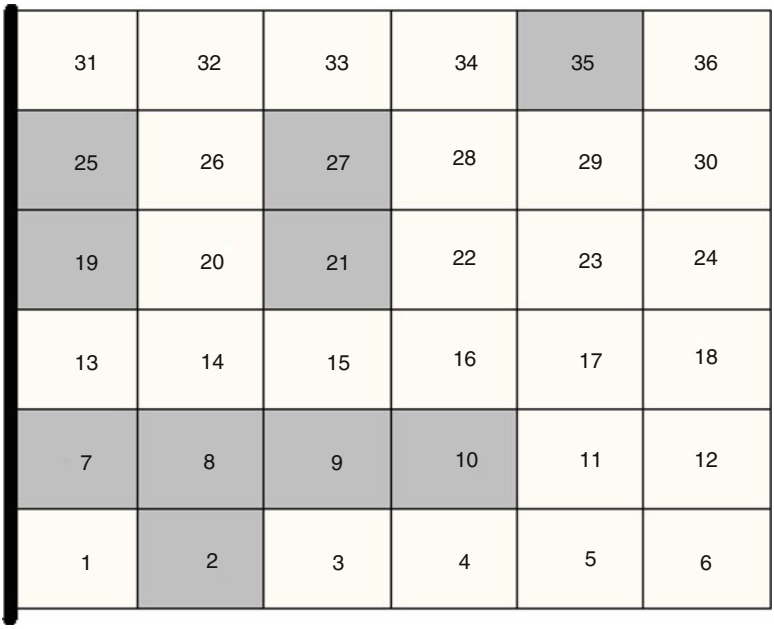


Fig. 2 Optimal actuator locations for the cantilever plate subject to a point force at the tip

optimal actuation voltages are close to the upper saturation limit. Figure 2 shows the optimal locations of the ten actuators on this plate when the presented hybrid GA is applied. The optimal value of the cost function is 3.983514e−03.

In Fig. 3, the achieved shape, when actuators are optimally located and the optimal voltages are applied on the plate, is compared with the desired position. By comparing the shapes in Fig. 3, it can be seen that the controlled deflection is quite close to the desired shape.

4.1.2 A Cantilever Plate Subject to a Uniform Load

In this example, the plate is originally flat and then is exposed to a uniform distributed load of $1\,\mathrm{Nm}^{-2}$. In order to determine the optimal placements of the piezoelectric actuators to flatten the plate, the GA is used with the following values: Generations = 1,000, Population = 50, and EliteCount = 2. The optimal values of the applied voltages as well as the optimal locations of actuators are given in Table 2 (see also Fig. 4).

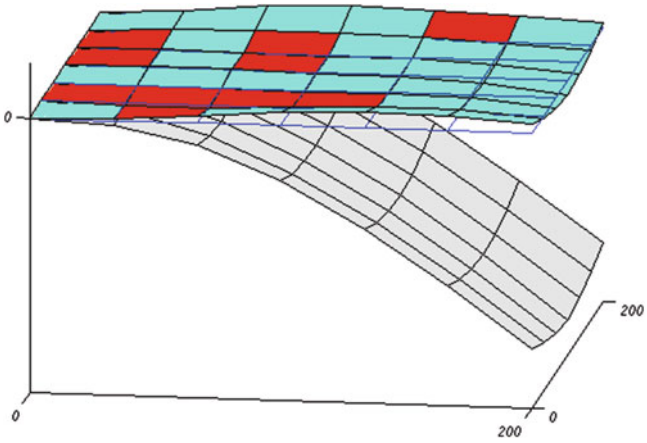


Fig. 3 Comparison between the achieved shape and the desired shape $w^d(x) = 0$ for a point load at the tip

Table 2 Optimal locations of ten actuators on the cantilever plate for the desired shape $w^d(x) = 0$

Optimal locations	Corresponding optimal voltages
1, 2, 5, 9, 10, 13, 19, 21, 22, 36	0.095, 2.994, 0.962, 0.307, 0.794, 2.198, 3.033, 1.412, 1.409, 0.003

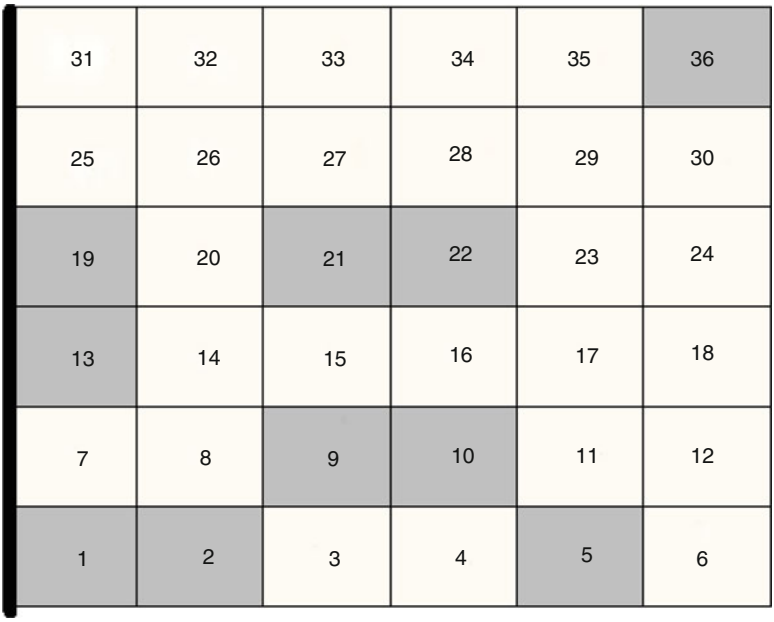


Fig. 4 Optimal actuator locations for the cantilever plate subject to a uniform load

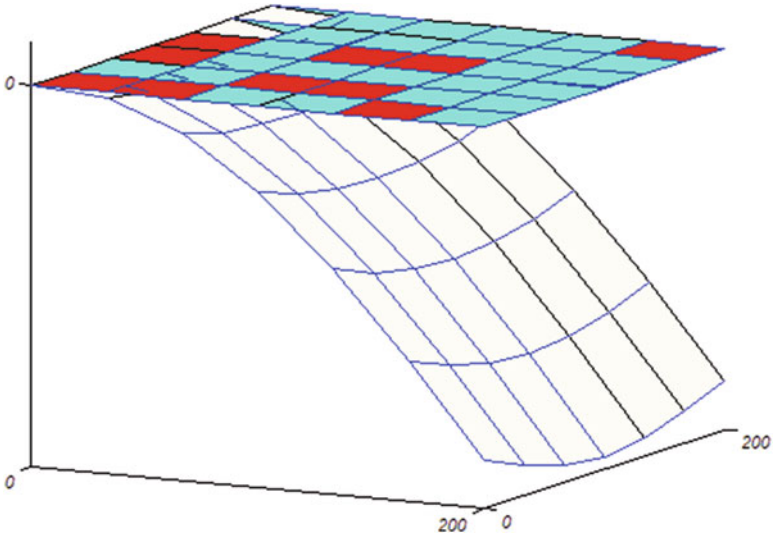


Fig. 5 Comparison between the achieved shape and the desired shape $w^d(x) = 0$ for uniform load condition

Genetic algorithm results in an objective value of 2.114123e-05. Figure 5 shows the achieved plate shapes from optimal placement. It should be noted that the values of the applied voltages are very small, showing that effective control can be achieved with suitable placement of actuators.

4.2 Optimal Actuator Locations in Vibration Control

Consider now the problem of optimizing the locations of piezoelectric patch actuators on a cantilever plate for active vibration control. In vibration control, the upper piezoceramics are severed as sensors and the lower ones as actuators. The first twelve modes are used in the modal space analysis and an initial modal damping ratio for each of the modes is assumed to be 0.8 %. The parameters taken for the genetic algorithm are: Generations = 100, Population = 100, EliteCount = 2. The negative of the controllability index in Eq. (14) has been taken as the fitness function. The optimal placement of actuators is shown in Fig. 6a.

To verify the optimization results, the plate is subjected to a vertical impulse at its tip and the disturbance in a structure is suppressed by using the LQR as a control measure with weight matrices $Q = 10^7 * I$ and $R = I$. The response at the free end of the plate is shown in Fig. 7 for the optimal placements that the genetic algorithm has developed. The responses from a non-optimal actuator placements are also included in the same figure. The nonoptimal placements are shown in Fig. 6b. Figure 7 shows that with the optimal placement the vibrations are suppressed much faster than the other nonoptimal locations.

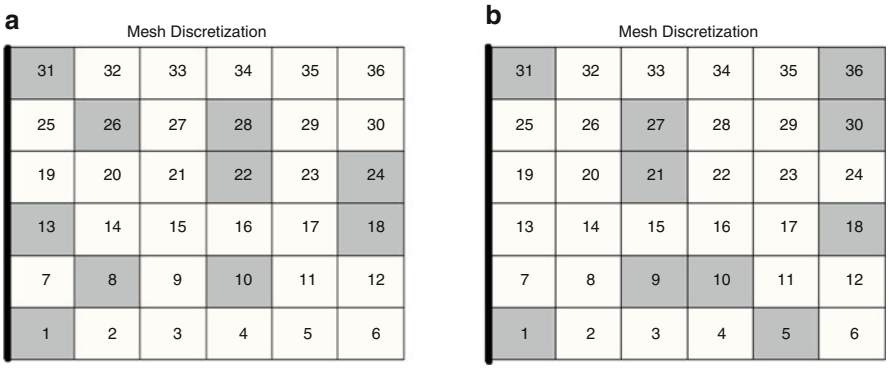


Fig. 6 (**a**) Optimal actuator location and (**b**) nonoptimal actuator location

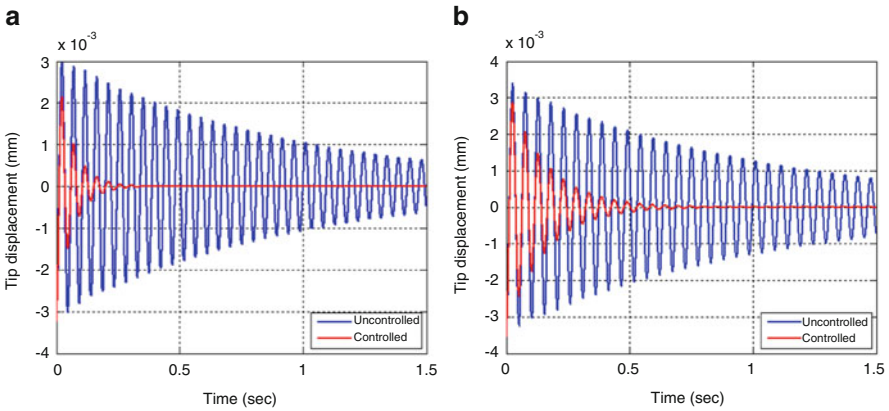


Fig. 7 Displacements at the plate tip: (**a**) optimal actuator location and (**b**) nonoptimal actuator location

5 Conclusions

Optimal design of smart structures leads to difficult optimization problems that must be solved numerically. Academic examples of optimal design in a piezoelectric controlled plate subjected to static and dynamic loadings have demonstrated that even with up-to-date general-purpose algorithms the calculation of global optimum is a challenge. In fact, optimal placements calculated in Fig. 4, for statics, and Fig. 6a, for dynamics, are not symmetric as expected. Nevertheless, the results are still useful for practical applications. Much more complicated problems arise in plate and shell structures with arbitrary shape and/or usage of more sophisticated design criteria.

Acknowledgements This research has been co-financed by the European Union (European Social Fund—ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)—Research Funding Program: ARCHIMEDES III. Investing in knowledge society through the European Social Fund. The authors gratefully acknowledge this support.

References

1. Chee, C., Tong, L., Steven, G.: Piezoelectric actuator orientation optimization for static shape control of composite plates. Compos. Struct. **55**(2), 169–184 (2002)
2. da Mota Silva, S., Ribeiro, R., Rodrigues, J.D., Vaz, M., Monteiro, J.: The application of genetic algorithms for shape control with piezoelectric patches: an experimental comparison. Smart Mater. Struct. **13**(1), 220 (2004)
3. Dueck, G.: New optimization heuristics: the great deluge algorithm and the record-to-record travel. J. Comput. Phys. **104**(1), 86–92 (1993)
4. Foutsitzi, G.A., Gogos, C.G., Hadjigeorgiou, E.P., Stavroulakis, G.E.: Actuator location and voltages optimization for shape control of smart beams using genetic algorithms. Actuators **2**(4), 111–128 (2013). doi:10.3390/act2040111
5. Gen, M., Cheng, R.: Genetic Algorithms and Engineering Optimization, vol. 7. Wiley, New York (2000)
6. Gupta, V., Sharma, M., Thakur, N.: Optimization criteria for optimal placement of piezoelectric sensors and actuators on a smart structure: a technical review. J. Intell. Mater. Syst. Struct. **21**(12), 1227–1243 (2010)
7. Hadjigeorgiou, E., Stavroulakis, G., Massalas, C.: Shape control and damage identification of beams using piezoelectric actuation and genetic optimization. Int. J. Eng. Sci. **44**(7), 409–421 (2006)
8. Han, J.H., Lee, I.: Optimal placement of piezoelectric sensors and actuators for vibration control of a composite plate using genetic algorithms. Smart Mater. Struct. **8**(2), 257 (1999)
9. Irschik, H.: A review on static and dynamic shape control of structures by piezoelectric actuation. Eng. Struct. **24**(1), 5–11 (2002)
10. Kumar, K.R., Narayanan, S.: Active vibration control of beams with optimal placement of piezoelectric sensor/actuator pairs. Smart Mater. Struct. **17**(5), 055008 (2008)
11. Lam, K., Peng, X., Liu, G., Reddy, J.: A finite-element model for piezoelectric composite laminates. Smart Mater. Struct. **6**(5), 583 (1997)
12. Rao, S.S., Pan, T.S., Venkayya, V.B.: Optimal placement of actuators in actively controlled structures using genetic algorithms. AIAA J. **29**(6), 942–943 (1991)
13. Wang, Q., Wang, C.M.: Optimal placement and size of piezoelectric patches on beams from the controllability perspective. Smart Mater. Struct. **9**(4), 558 (2000). http://www.stacks.iop.org/0964-1726/9/i=4/a=320

Stable EEG Features

V. Stefanidis, G. Anogiannakis, A. Evangelou, and M. Poulos

Abstract The aim of this chapter is to identify stable points and stationary wavelets in EEG signals. Generally an EEG signal is a very complex nonstationary signal. It is very difficult to recognize specific EEG features such as Biometric patterns and Pathological changes. Using a repeated autocorrelation procedure and symmetry features of EEG time series on real EEG Time Series Data, we experimentally investigate stable points in EEG signals. Also we investigate standing waves shafts around these stable points, which reveals the existence of stationary wavelets in EEG signals.

Keywords EEG signal • Data mining • Stationarity • Time series • Autocorrelation coefficients • LVQ neural network

1 Introduction

Time-frequency analysis of electroencephalogram (EEG) through several mental tasks received significant consideration. As EEG is nonstationary, time-frequency analysis is crucial to analyze brain states during different mental tasks [1]. In particular, it has been exposed that large-scale patterns of matched neuronal EEG activity are ever varying and thus exhibit a substantial variability over time. Therefore, until now, analysis of the EEG signal has been based mostly on statistical data processing in order to acquire stable and reliable characteristics. The assumption underlying

V. Stefanidis • A. Evangelou
Laboratory of Physiology, Medical School, University of Ioannina, Greece
e-mail: vstefan@cc.uoi.gr; aevaggel@cc.uoi.gr

G. Anogiannakis
Laboratory of Physiology, Medical School, Aristotle University, Thessaloniki, Greece
e-mail: anogian@auth.gr

M. Poulos (✉)
Laboratory of Information Technologies, Faculty of Information Science and Informatics, Ionian University, Corfu, Greece
e-mail: mpoulos@ionio.gr

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_18

such statistical analyses is the "stationarity" of the registered signal [1]. However, in studies [1, 2], the EEG sources are considered quasi-stationary. In study [3] is introduced that the length of EEG of which is so short that signal within it can be treated as stationary or quasi-stationary [3].

The basic aim of the research study is to examine the stationary lengths of each EEG in order to be corroborated the hypothesis of the previous study [3]. Thus, for this implementation a novel EEG signal analysis is introduced.

It is known that the time-frequency information of EEG signal can be used as a feature for classification in brain–computer interface (BCI) applications [4–14] or for Diagnostic Purpose [7].

There are two alternative approaches to examining stationarity, the parametric and the nonparametric. Parametric approaches are widely used by those undertaking research in the time domain, such as economists, who are building certain assumptions about the nature of their data.

Nonparametric approaches are most commonly seen by researchers working in the frequency domain, such as electrical engineers, who often treat the system as an unknown entity and cannot make any inferences or reckoning based on the nature of the data.

Nonparametric tests are not based on the knowledge that the population data are normally distributed. By making no assumptions about the nature of the data, non parametric tests are more widely applicable than parametric tests which often require normality in the data.

While more widely applicable, the downside is that nonparametric tests are also less powerful than parametric tests merely because the assumptions underlying their use are fewer and weaker than those associated with parametric tests.

In this study, the symmetrical features of EEG are investigated using a well-fitted autocorrelation coefficients (ACC) procedure. For this the schedule of this algorithm is developed in the following three steps:

1. The ACC calculation using symmetrical features
2. Polynomial procedure on ACC
3. Graph Interpolation Procedure

2 The Method

2.1 The ACC Algorithm

This study is based on the hypothesis that the shape of a segment of a Time Series may be described by the degree of asymmetry around a characteristic point. The degree of asymmetry of a segment is obtained via the Pearson criterion [9] and is described by the following equation:

$$S = \frac{\overline{X} - M_o}{s} \tag{1}$$

where S is the degree of asymmetry, $\overline{X}$ is the mean value of a time series segment, M_o is the value of the characteristic time series (data) point, which is received as the maximum value, s is the standard deviation of a time series segment. The degree of asymmetry may be characterized as a necessary characteristic coefficient of the time series segment in our case because this depicts a total geometry picture of the segment. In this stage, the time series segment x_t is subjected to power-spectral-density analysis of each time series overlap segment, and computed using frequency estimation of the standard periodogram, as follows:

$$S_x(f) = \frac{1}{T}|\sum_{t=1}^{T} x_t e^{j-2\pi tf}|^2 \tag{2}$$

The samples x_t are replaced by the values of the periodogram given by $|f_n|$ where, which can be computed using the fast Fourier transform (FFT) algorithm [10] thus, according to the Inverse Function of formula 1, which is given as follows:

$$f_n = (S_n^{-1}(S_n f)) \tag{3}$$

This approach has also been applied to several cases, e.g., EEG study [4–14]. Using Eqs. (2) and (3), Eq. (1) is transformed into 4:

$$S_k = \frac{T\sqrt{T-2}(|\bar{\hat{f}}| - |f_g|)}{\sqrt{2T\sum_{k=1}^{T/2} f_n^2 - 4(\sum_{k=1}^{T/2} f_n)^2}} \tag{4}$$

where $f_g = max(f_n)$ for $1 \leqslant g \leqslant (int[T/2])$.

Thereafter, we considered set D of sequences, which consists of the following coefficients:

$$\{D\} = \{\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_{k-1}\}$$

where

$$\hat{D}_1 = \{S_1, S_2\}, \hat{D}_2 = \{S_1, S_2, S_3, \}$$
$$\hat{D}_3 = \{S_1, S_2, S_3, S_4\}, \ldots, \hat{D}_{k-1} = \{S_1, S_2, S_3, \ldots, S_k\}$$

Then, the ACC of the proposed method are computed as follows:

$$\hat{C} = [C_m] \tag{5}$$

where $m = 3, \ldots, k+2$ is calculated via the condition of Eq. (6) which expresses the corresponding weighing function using a Power Spectral Density window:

$$w_m = int(\frac{T}{m}), \text{ where } w_m \geqslant 3 \tag{6}$$

Then, the ACC are given by

$$S_k = \sqrt{\frac{2k\sum_{m=1}^{k+2}(\hat{D}_k\hat{D}_k^T) - 4 \mid \sum_{m=1}^{k+2}(\hat{D}_k) \mid^2}{m(m-2)}} \tag{7}$$

2.2 Polynomial Procedure on ACC

After that, we have to find the coefficients of the sixth degree polynomial $p(x)$ (see [4–14], as a justification of this option of 6 degrees of freedom for the fitting), in order to achieve best fitting of $p(x(i))$ to $C(i)$, where $x(i)$ are the horizontal axis elements (time or points) and $C(i)$ are the ACC. The produced polynomial is known as "Interpolation Polynomial." The result p is a row vector of length $n+1$ containing the polynomial coefficients in descending powers:

$$p(x) = p_1x^n + p_2x^{n-1} + \ldots + p_nx + p_{n+1} \tag{8}$$

2.3 Interpolation Procedure

Then, in order to find exactly the graph which best fits the coefficients of the Interpolation Polynomial, for each element of the horizontal axis (x) we compute the corresponding element on the vertical axis (y) according to following equation:

$$y = p_1x^n + p_2x^{n-1} + \ldots + p_nx + p_{n+1} \tag{9}$$

where p_i is the previously computed Interpolation Polynomial. This is achieved using delta error estimation, in which delta is an estimate of the standard deviation of the error in predicting a future observation at x by $p(x)$ [see Eq. (8)]

3 Experimental Part

As input for our experiments we used EEG signals which we received from people who were in the process of relaxation. Three men and a woman participated in this procedure. The duration of the EEGs was approximately 46 s (46,000 ms). The input files approximately consist of 23,000 points (1 point/2 ms).

For the implementation of the experiments we made a function at Matlab, which had an input of two parameters. The first parameter (k) is the number of windows (samples) we will use from the EEG section, the second parameter (w) is the size of the window (sample length). In case the number of windows is more than one ($k>1$), we have overlapping of used points (for instance 5000…5020, 5001…5021, etc). We executed the experiments with several combinations of the

two parameters and we received several results which we present below. We used mostly samples (w) of 20 points. From such a sample, of 20 points length, 28 ACC $\{D\}$ are isolated.

As we said in the previous chapter, we have to compute the coefficients of the sixth degree "Interpolation Polynomial" $p(x)$ [Eq. (8)], in order to achieve best fitting of $p(x(i))$ to $C(i)$, where $x(i)$ are the horizontal axis elements (time or points) and $C(i)$ are the ACC. To achieve that we used the Matlab polyfit function:

$$p = polyfit(x, s, n) \tag{10}$$

Figure 1 we can see a plot of these coefficients.

Then, in order to find exactly the graph which best fits the coefficients of the Interpolation Polynomial, for each element of the horizontal axis (x) we compute the corresponding element on the vertical axis (y) according to following equation:

$$y = p_1 x^n + p_2 x^{n-1} + \ldots + p_n x + p_{n+1}$$

where p_i is the previously computed Interpolation Polynomial.

For every value x of the horizontal axis (time or points), the polyval function of Matlab returns the value of a polynomial of degree n evaluated at x:

$$p = polyval(p, x) \tag{11}$$

In Fig. 2 we can see the graph produced by the procedure described above.

The diagram we see in Fig. 2 corresponds to the representation of the polynomial [Eq. (9)] which is the result of the execution of the function with $k = 1$ and $w = 20$ for a "man" subject. In Fig. 3 we can see a similar representation of the same polynomial of a woman and another man EEG.

If we increase the number of samples $k = 10$ we get the representation of Fig. 4. The representations we get, as we said before, have overlapped points.

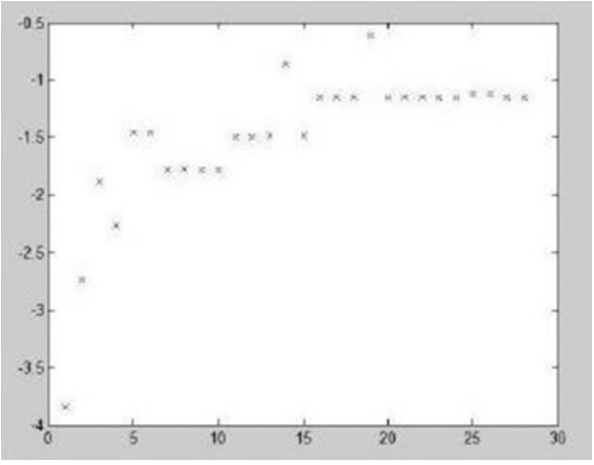


Fig. 1 Coefficients of interpolation polynomial

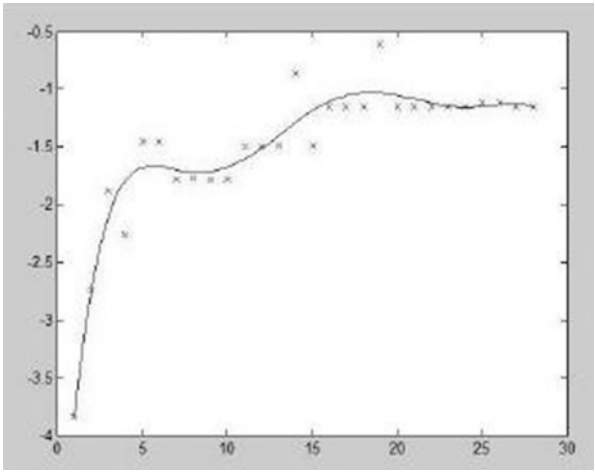


Fig. 2 Polynomial of degree n evaluated at x [Eq. (9)] graphical representation

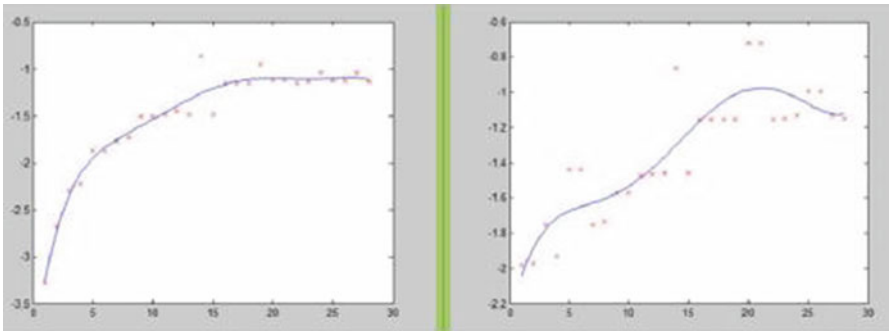


Fig. 3 Polynomial [Eq. (9)] representation for a woman and a man, respectively

In Fig. 4 also, we can clearly distinguished the points we have stationarity. We can easily distinguish these stable points (points with approximate zero amplitude of motion) among which standing wave shafts are formed.

At the female signal we were forced to increase the width of the sample in order to find the points for which stationarity appears. In Fig. 5, in the first diagram k is equal to 10 ($k = 10$), w is equal to 20 ($\mathrm{w} = 20$) and as signal section, we received, randomly, the points from 5,000 up to 5,500. In the first diagram the stable points are not easily distinguished. In the second diagram of the same figure k is equal to 10, w is equal to 20, and as a signal section, we received the points from 3,000 up to 5,500 (we increased the section length). In this figure the stable points as well as the shafts of standing waves are easily distinguished (Fig. 5).

A standing wave pattern is not actually a wave, but rather a pattern of a wave. Thus, it does not consist of crests and troughs, but rather nodes and antinodes. In other words, shafts are formed. The pattern is the result of the interference of two waves to produce these nodes and antinodes. The parameters of this pattern are shown in Fig. 6. In Fig. 7 we can see 6 EEG's per person which depicted with a

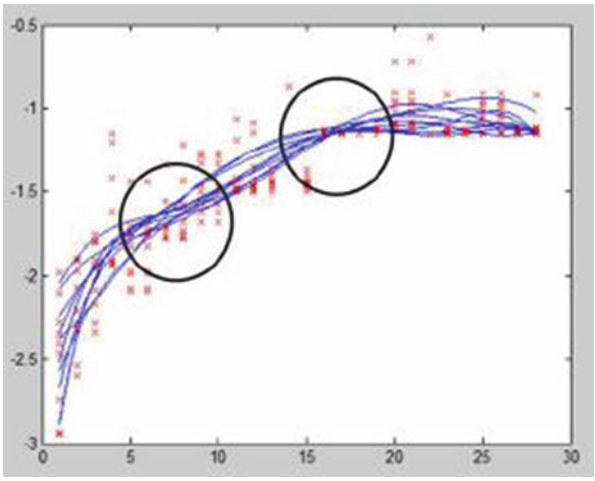


Fig. 4 Polynomial [Eq. (9] representation for a man ($k = 10$,$w = 20$), stable points

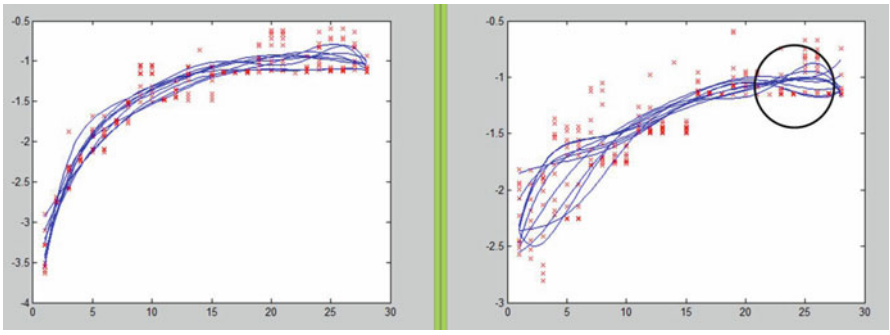


Fig. 5 Polynomial [Eq. (9)] representation for a woman (5,000–5,500 and 3,000–5,500 points)

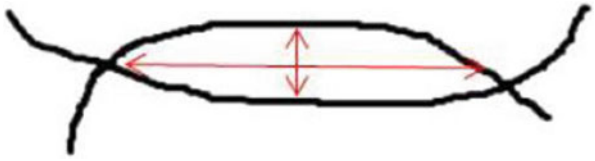


Fig. 6 Static waves pattern parameters (wavelength and amplitude of oscillation)

color and (total 5 persons). Each EEG is submitted (auto repeat 20 times). In this figure we can have a more cautious approach of standing waves.

4 Conclusions

According to above findings we conclude that stable points are presented clearly during many intersections of the polynomials. These findings are consistent with the theory [15] in which a common stable point may be shown to exist then

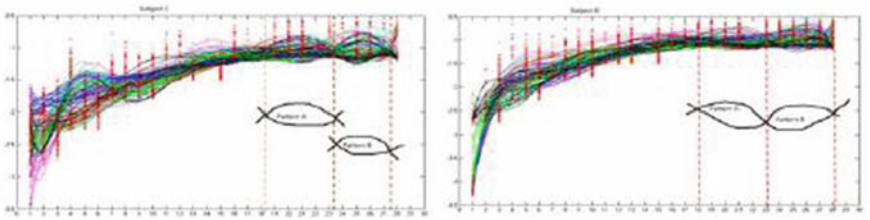


Fig. 7 Static waves in 6 EEG's per person

the commuting polynomials have a common stable point and the community polynomials yield a certain homomorphisms [16]. However, the homomorphisms are continuous functions that preserve topological properties [17] and these are very significant for pattern recognition reasons such as biometric EEG features [18] and for diagnostic purposes [19].

Furthermore, the Data Mining of Time Series using Autocorrelation Coefficients and symmetry features of EEG time series is addressed in this work. A repeated autocorrelation procedure was performed on real EEG Time Series Data, in an attempt to experimentally investigate and establish the connection between Time Series data and hidden information relating to the properties of stationary Time Series. These results are in agreement with previously proposed research methodology/methods, exhibiting a Time-Series Analysis of real EEG, which carries genetic information, as well as demonstrating the potential relevance of our approach for Stationary Identification as a tool in Time Series Analysis.

Results show that the proposed algorithm can provide an optimal time-frequency resolution using autocorrelation procedure in order to highlight stable EEG points around a trust region. Although it is generally accepted that the EEG signals are not stationary, we could isolate points which are stable and we can distinguish stationarity.

Our future research will be focused on the following points:

- The verification of these results with more EEG data is the next step of the proposed research.
- More extensive experimentation is needed in order to obtain statistically significant results and, thus, check and verify assumptions on a Real Data set about the existence of a one-to-one correspondence between the Time Series and symmetric spectral features.
- The wavelet pattern recognition in order to create personal wavelet is the next aim.
- The comparison of the verification of the unique feature of each wavelet using neural network is asked in the third researcher step.
- Finally, the connection of the findings and the possible biological-biometric features are considered as further target of this research.

References

1. Kaplan, A.Y., Fingelkurts, A.A., Fingelkurts, A.A., Borisov, S.V., Darkhovsky, B.S.: Nonstationary nature of the brain activity as revealed by EEG/MEG: methodological, practical and conceptual challenges. Signal Process. **85**(11), 2190–2212 (2005)
2. Zygierewicz, J., Mazurkiewicz, J., Durka, P.J., Franaszczuk, P.J., Crone, N.E.: Estimation of short-time cross-correlation between frequency bands of event related EEG. J. Neurosci. Methods **157**(2), 294–302 (2006)
3. Wang, C., Xu, J., Lou, W., Zhao, S.: Dynamic information flow analysis in vascular dementia patients during the performance of a visual oddball task. Neurosci. Lett. **580**, 108–113 (2014)
4. Poulos, M., Rangoussi, M., Alexandris, N.: Neural network based person identification using EEG features. In: Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999, pp. 1117–1120. IEEE (1999)
5. Poulos, M., Rangoussi, M., Chrissikopoulos, V., Evangelou, A.: Parametric person identification from the EEG using computational geometry. In: Proceedings of ICECS'99. The 6th IEEE International Conference on Electronics, Circuits and Systems, 1999, pp. 1005–1008. IEEE (1999)
6. Poulos, M., Rangoussi, M., Chrissikopoulos, V., Evangelou, A.: Person identification based on parametric processing of the EEG. In: Proceedings of ICECS'99. The 6th IEEE International Conference on Electronics, Circuits and Systems, 1999, pp. 283–286. IEEE (1999)
7. Poulos, M., Georgiacodis, F., Chrissikopoulos, V., Evagelou, A.: Diagnostic test for the discrimination between interictal epileptic and non-epileptic pathological EEG events using auto-cross-correlation methods. Am. J. Electroneurodiagnostic Technol. **43**, 228 (2003)
8. Poulos, M., Rangoussi, M., Chrissikopoulos, V., Evangelou, A., Georgiacodis, F.: Comparative analysis of the computational geometry and neural network classification methods for person identification purposes via the EEG: part 1. J. Discret. Math. Sci. Cryptogr. **7**, 319–347 (2004)
9. Poulos, M., Papavlasopoulos, S.: Automatic stationary detection of time series using auto-correlation coefficients and LVQ - Neural network. In: Fourth International Conference on Information, Intelligence, Systems and Applications (IISA), 2013, pp. 1–4. IEEE (2013)
10. Mendonca, M.W.: Multilevel Optimization: convergence theory, algorithms and application to derivative-free optimization. Ph.D. thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium (2009)
11. McArdle, J.J., Hamagami, F.: Modeling incomplete longitudinal and cross-sectional data using latent growth structural models. Exp. Aging Res. **18**, 145–166 (1992)
12. Felzer, T.: On the possibility of developing a brain-computer interface (bci). Technical University of Darmstadt, Department of Computer Science, Darmstadt (2001)
13. Palaniappan, R.: Brain computer interface design using band powers extracted during mental tasks. In: Conference Proceedings of 2nd International IEEE EMBS Conference on Neural Engineering, 2005, pp. 321–324. IEEE (2005)
14. Marcel, S., Millán, J.R.: Person authentication using brainwaves (EEG) and maximum a posteriori model adaptation. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 743–752 (2007)
15. Boyce, W.M.: Commuting functions with no common fixed point. Trans. Am. Math. Soc. **137**, 77–92 (1969)
16. Ritt, J.F.: Permutable rational functions. Trans. Am. Math. Soc. **25**, 399–448 (1923)
17. Oxtoby, J.C., Ulam, S.M.: Measure-preserving homeomorphisms and metrical transitivity. Ann. Math. **42**, 874–920 (1941)
18. Poulos, M.: On the use of EEG features towards person identification via neural networks. Inform. Health Soc. Care **26**(1), 35–48 (2001)
19. Poulos, M., Papavlasopoulos, S., Alexandris, N., Vlachos, E.: Comparison between auto-cross-correlation coefficients and coherence methods applied to the EEG for diagnostic purposes. Med. Sci. Monit. **10**(10), 99–100 (2004)

Deriving Pandemic Disease Mitigation Strategies by Mining Social Contact Networks

M. Ventresca, A. Szatan, B. Say, and D. Aleman

Abstract In this chapter we propose a robust approach to deriving disease mitigation strategies from contact networks that are generated from publicly available census data. The goal is to provide public policy makers additional information concerning the type of people they should aim to target vaccination, quarantine, or isolation measures towards. We focus on pandemic disease mitigation, but the approach can be applied to other domains, such as bioterrorism. The approach begins by constructing a representative contact network for the geographic area (we use the Greater Toronto Area of ≈ 5.5 million individuals) from census information. Then, network centrality measures are employed to ascertain the importance of each individual to the proper topological functioning of the network. The top-ranked individuals' characteristics, as defined by census information, are then used as input to decision tree classifiers. The resulting output is a set of high-level rules that identify potential types of individuals to target in order to mitigate disease spread. Experimental evidence for the efficacy of the approach is also provided.

Keywords Pandemic planning • Contact network • Decision tree • Public policy generation

1 Introduction

Estimates of the potential impact of a pandemic disease range to upwards of hundreds of millions of global deaths and trillions of dollars in socioeconomic costs [36, 48]. While immunization remains the preferred strategy, during the early onset

M. Ventresca (✉)
School of Industrial Engineering, Purdue University, West Lafayette, IN, USA
e-mail: mventresca@purdue.edu

A. Szatan • B. Say • D. Aleman
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada
e-mail: aleman@mie.utoronto.ca

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_19

of a pandemic vaccines are unlikely to be available or will require many months to produce en masse [46]. For diseases that spread through social contact, early public policy intervention strategies concentrate on social distancing. However, if the strategies are too burdensome they will needlessly disrupt socioeconomic circumstances [42, 43, 62]. Thus, the development of high quality public policies for mitigating pandemic diseases is of critical importance [7, 10, 11, 19, 23, 31, 47, 50, 51, 57].

In this chapter we investigate the implied public policy decisions that result from using network centrality measures to identify a subset of individuals to target public policies towards. We construct our social contact network from publicly available census and travel information of the Greater Toronto Area (GTA) and perform subsequent decision tree mining of the population constituents using information such as age and travel distance to work. Each branch of the resulting decision tree implies a particular subset of individuals to target, to which a number of corresponding public policies may exist. The resulting decision tree will correspond to a number of potential strategies for containing the disease.

One traditional approach to devising better policies is by examining past policy decisions and their observed consequences, within the demographic context at that time. Another approach focuses on simulating of "what if" scenarios and policies through the use of a variety of computational approaches. Whatever the decided policies to implement, it is unlikely to observe a low public compliance unless the policies are overly disruptive to their income or career [15]. Consequently, low-income families may have a tougher time than others. This also suggests that communities should be in continuous preparation for mitigating an outbreak. As indicated in [14], businesses will play a large role in minimizing these issues through programs such as working from home and paid sick leave. The ability to discover quality mitigation policies can be greatly improved if subsets of individuals to target can be accurately and efficiently identified, which is the goal of this work.

The problem of developing a large-scale and effective pandemic mitigation strategy involves a number of considerations, and sometimes conflicting studies to consider. Nevertheless, mathematical models can be an invaluable tool for uncovering potential implications of disease transmission characteristics and the impact of public policy decisions on mitigation. Traditional epidemiology models of disease spread assume homogeneous and random mixing of individuals and are governed by differential equations [4, 37]. The assumption of homogeneity abstracts away much of the epidemiologically important sources of variability, such as age, sex, contact rate, and compliance to public health recommendations [8]. This individual-level diversity can have profound effects on the population level disease dynamics, and as such there has been a recent trend towards network-based heterogeneous models [2, 5, 8, 24, 27, 49, 59].

Using a large-scale agent-based simulation model [28] studied contact networks representing the city of Portland. They considered the impact of different vaccination strategies based on network structure and described how to construct the contact network from survey, census, and transportation data. In a subsequent study [29], the question of whether sufficient detail is included in a contact network

to adequately represent the region's contact network is discussed with a statistical test being proposed. Salathe et al. [50] focus on controlling diseases on networks having community structure. They find that community structure strongly affects disease dynamics, and develop an efficient algorithm for identifying individuals for targeted vaccination. A method for optimizing disease interventions during an outbreak was proposed in [61] that uses information available during the early stages of a pandemic. The approach provides rules to determine which measure should be taken given observed disease characteristics. If groups are appropriately identified in the population then it is possible to achieve high quality interventions.

Recent studies suggest that a large amount of the variation in simulated endemic disease levels can be described by structural properties of mean degree, clustering coefficient, and average path length measures of graphs having exponential degree distributions [3]. Unfortunately, the latter metric requires significant computational overhead for large graphs, but achieving high levels of accuracy is attainable using only the former two measures, which are much more computationally feasible for the very large graphs considered here. In comparison, network-based simulations require enormous amounts of computational effort (usually many hours or days) and have high computer memory requirements [9, 18, 21, 25, 27, 56].

The rest of this chapter is organized as follows. Section 2 describes how the representative population is constructed from census data, and how mixing patterns between individuals are generated. The approaches to identifying a subset of individuals who play an important role in the proper functioning of the contact network are outlined in Sect. 3, as are decision trees. Experimental results that highlight a sample of decision trees and their implied public policy targets are presented in Sect. 4. The final section provides a brief discussion of the results and directions for future work.

2 The Contact Network

Contact networks can be thought of as a graphical representation of social interactions over a fixed period of time. Individuals of the population are the nodes of the network and interactions between them are the edges. Often edges are weighted to correspond to the strength of interaction between pairs of individuals. In the context of pandemic disease mitigation edge weights are typically related to the likelihood that the disease will transmit between the individuals [34]. An important assumption underlies the manner in which the edges are constructed. Specifically, it is assumed that edges accurately represent the method of disease transmission. That is to say, a social contact network is not likely the best model for understanding a disease that is spread through drinking water. Throughout this work we assume that avenues for transmission are well represented by a social network. We also assume a standard SIR-based model whereby all individuals are initially susceptible to the disease and can only be infected once.

In this study we concentrate on the GTA, which has a population of approximately 5.5 million people. Each individual in the contact network is generated with characteristics representative of those reported in publicly available census information [53] such as age, type of career, and approximate home and work address. For privacy reasons only marginal distributions per characteristic are made available to the public. However, a number of reasonable assumptions can be made to enforce the construction of more realistic contact structures, as described below.

2.1 Generating a Representative Population

The GTA is partitioned into a number of adjacent units called dissemination blocks, which are the smallest grouping by which census data are reported. Typically, these blocks are bounded by roadways and may correspond to city blocks in urban areas or represent a group of adjacent rural blocks in sparsely populated areas. The GTA contains 7,684 dissemination blocks and each individual lives in exactly one of them.

The census includes a number personal characteristics such as age and family structure and the type of occupation as listed in Table 1. Despite being reported as marginal distributions, a number of societal structures can be safely assumed in order to construct a contact network from this information. For instance, it is safe to assume that each household must contain at least one adult (age ≥ 18). If other adults are present they are most likely to be a spouse, but in some cases children live with grandparents or parents are near retirement and their children are young adults who live at home. In order to model these different situations we presume that a parent or legal guardian of a child must be at least 18 years older than the child. Using the statistics of the census data we can regenerate the exact number of total households with an appropriate family structure and age distribution.

It is also important to accurately model the major public transportation routes given the role they can play in disease spread [54]. While mitigation strategies focused on limiting public transportation use are cumbersome, they have been shown as measures that most individuals would strongly consider during a pandemic [39]. The GTA contains four subway routes and sixteen major train and bus lines [55]. Usage statistics are publicly available per stop along a route [55], which are used in conjunction with census information (i.e., home and work dissemination blocks) to assign individuals to a particular route, representing their journey to and from work.

Children aged 4–18 are all assumed to attend an educational institution. However, it cannot be assumed that children attend a school within their dissemination block since a number of rural areas lack the sufficient number of children to support a public school. Moreover, a number of schools may exist in a particular dissemination block, or an individual may live near the border of a dissemination block with a school in an adjacent dissemination block. To resolve this issue a partitioning of the GTA is performed using a Veroni tessellation [6] such that

Table 1 Characteristics for each individual of the population as used by the Canadian census [53]

Attribute	Description
Age	Uniformly distributed over each of 18 subintervals of $[0, 100)$
Home location	One of the 7,684 dissemination blocks
Household size	$[1, \ldots, 6+]$
Family structure	{Lone parent, couple, other structure}
Public transportation	{Not taken, subway, public transit}
Transit route	If public transportation is used, one of $[0, \ldots, 19]$ routes
Worker type	{Skilled trades, office/clerical, high volume, low volume, other}
Workplace location	If applicable, one of the 7,684 dissemination blocks
Distance to work	Distance $(0, \infty)$ in meters between home and work dissemination blocks
Commuter	Whether working individuals commute to work or not
Office size	If applicable, $[1, 500]$
School attended	If applicable, one of 2,232 schools
School size	If applicable, maximum school size is 5,000 students
Health network	One of 5 health networks that cover all health-based services
Clinician type	If applicable, {nurse, doctor}
Patient	Whether the individual is hospitalized or not
Hospital	If applicable, one of 50 local hospitals (as patient or employee)

Corresponding reported statistics are used to create the population

each resulting area contains exactly one school. We then assume students attend schools closer to their home, and assign them accordingly. If the closest school becomes full during the assignment process, the student is assigned to the next closest institution, and so on. Students' individual grade levels are also generated to accurately represent the information provided in the census.

Working-age adults (age 18–64) inform the census bureau of the type of job they work in, classified as skilled trades, office/clerical, high volume service, low volume service or other, in addition to clinicians (both doctors and nurses). Office size and locations are also publicly available and individuals are assigned to work in a particular area/office based on their job-related census information. Hospitals are treated as an office place with specialized employees. Retired individuals (age ≥ 65) are considered roaming members of the community and not assigned to any school or workplace.

2.2 Generating Appropriate Contact Patterns

After generating a representative population of individuals and assigning them to households, schools, workplaces, and transportation routes, the network of social interactions must be created, weighted by the strength of each interaction. In general, the connection strength between individuals i and j will be approximated by

a combination of contact duration t_{ij} (in minutes) and the likelihood, per unit time, of disease transmission occurring s_{ij}. That is, edge weights w_{ij} can be computed as $w_{ij} = t_{ij}s_{ij}$.

Methods for accurately quantifying the likelihood of disease transmission between individuals are the subject of current investigations, for instance [12, 22, 52]. As a simplification we let $T \in [0,1]$ represent an upper bound on the expected transmission likelihood between any two individuals, per unit time. This parameter depends on the disease being considered and allows us to simplify the edge weights to $w_{ij} = t_{ij}$. This contact duration is uniformly generated and tuned for the GTA network based on contact matrices in literature [2, 26, 35, 45, 59, 60]. Below we indicate the manner in which individuals are connected within each environment.

Households: A common assumption is that members of the same household will have a relatively large total contact duration. Hence, we model connections within a household as a complete graph. Edge weights are chosen according to a uniform probability distribution over the interval $[1,800]$, where the age of the two interacting individuals is used to narrow the range to more realistic values. For instance, a randomly selected infant (age < 4 years) will have a higher contact, on average, with adults than a random adult will have with an infant. The nonsymmetric 4×4 contact duration matrix represents the interaction between individuals of age groups $\{[0,4],[5,18],[19,64],[65,99^{+}]\}$ and closely resembles that of [26, 27].

Workplaces: The number of workplace contacts and their duration are affected by the type of business and number of employees. We utilize a Poisson-based sampling strategy where the rate parameter $\lambda = 7$ is the expected number of contacts the individual in question is estimated to make during a random day with fellow employees. Individuals who work in high volume positions will make an additional number of contacts as determined according to a uniform distribution over the interval $[10,80]$, with contact duration between $[1,8]$ minutes per interaction.

Schools: As children age the total duration spent with classmates will often exceed the total duration spent with family members [34]. The number of connections is determined according to the Poisson sampling approach used for workplaces, but with $\lambda = 8$. The individuals who make interactions are chosen randomly from among the individuals in their school and grade level. Teachers are assumed to make contact with half of their class, on average. It should be noted that the Poisson connection strategy is an approximation that is unlikely to accurately reproduce community structures.

Public transportation: Public transportation presents an interesting challenge due to the large number of people who typically utilize such services and the problem of assigning individuals to transportation routes given their geographic constraints. We assign individuals to one of the 20 transportation routes based on daily usage statistics at each station and the location of their home and workplace. The duration of contact in the public transportation system is modeled similarly to that of high volume work places, but over the range $[1,60]$ minutes. The number of interactions with other travelers also uses a Poisson sampler with rate tuned to $\lambda = 10$.

Community contacts: Random contact with individuals in the larger community is an important type of interaction to model [1, 34]. These contacts are made between two random people, where the contact duration is chosen in a similar fashion as for public transportation but at a lower connection rate of $\lambda = 5$ and maximum contact duration based on the age of the individuals taking part in the interaction.

2.3 Resulting Network Structure

The GTA model contains precisely 5,476,158 individuals, and approximately 110 million edges are created through the aforementioned connection process. This amounts to a mean vertex degree of approximately 40 and standard deviation of typically between 43 and 47. Validation of the network structure is performed by comparing the resulting mixing patterns to those of known contact network models (that were based on other urban areas/census data) and allowing for minor differences in resulting demographics. Specifically, we compared the total number of connections and the average duration of contact between all pairs of ages, in addition to vertex degree and edge weight distributions, to those reported in [26, 28, 33, 44, 60]. It should be noted that while results are given for a single generated network, the model generates statistically similar networks. Hence, the connection patterns and statistics between generated networks are likely to have negligible differences. Figure 1 shows the total number of contacts made between individuals. Figure 2 depicts the degree distribution on a log–log scale and a histogram of contact duration.

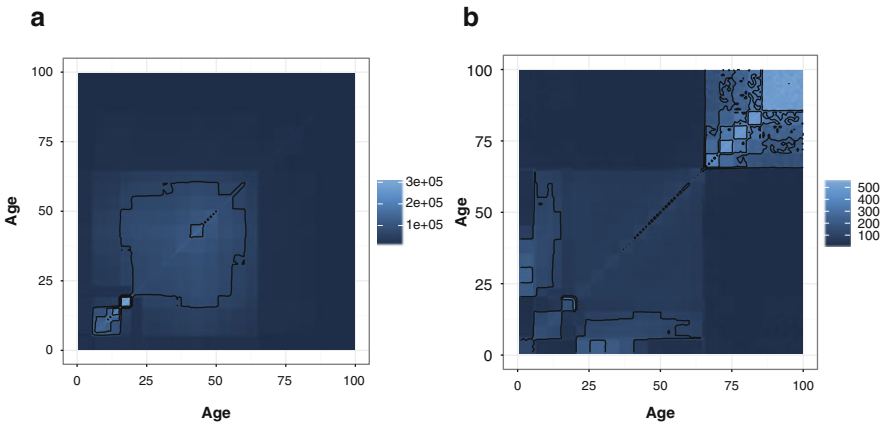


Fig. 1 Mixing pattern between ages according to the (**a**) total number of contacts made and (**b**) the average duration of contact

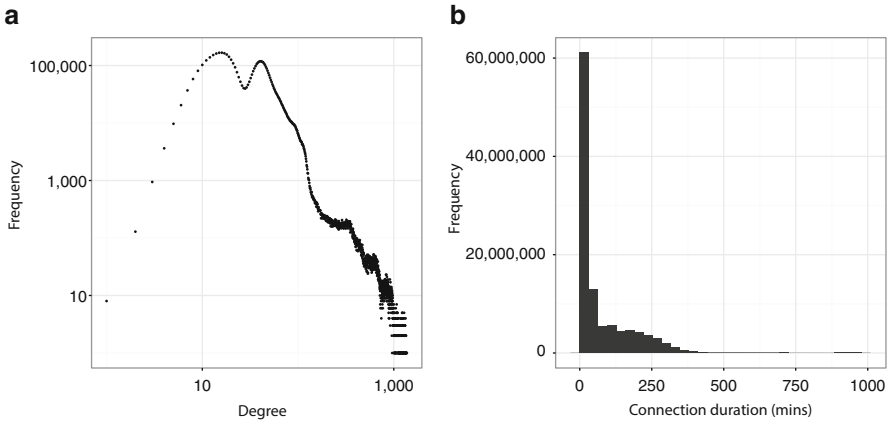


Fig. 2 Network (**a**) degree distribution (log-log scale) and (**b**) histogram of contact durations

2.4 Calculating $\mathcal{R}_0$ from the Network Structure

The basic reproduction number $\mathcal{R}_0$ quantifies the number of secondary infections caused, on average, over the time an individual is infectious. It is commonly utilized to quantify the severity of a pandemic. The pandemic will likely die out if $\mathcal{R}_0 < 1$ or become endemic if $\mathcal{R}_0 > 1$. A number of methods for accurately estimating $\mathcal{R}_0$ have been proposed [13, 20, 30, 32], although issues in its estimation and interpretation still remain [41].

From the network perspective, computing $\mathcal{R}_0$ can be accomplished using the technique of Newman [45]:

$$\mathcal{R}_0 = T\left(\frac{\sum_{d=1}^{\infty} p_d d^2}{\sum_{d=1}^{\infty} p_d d} - 1\right) = T\left(\frac{\langle d^2\rangle}{\langle d\rangle} - 1\right) \tag{1}$$

where p_d is the observed probability of degree d, T is the disease transmissibility (we assume $T = 1$ unless otherwise noted), and $\langle d\rangle$ and $\langle d^2\rangle$ are the mean degree and mean squared degree of the network, respectively. Thus, removing vertices or edges from the network through mitigation strategies will impact the values of $\langle d\rangle$ and $\langle d^2\rangle$, and thus the prevalence of the disease.

2.5 Calculating the Probability of a Pandemic

It is also possible to estimate the probability of pandemic occurring from an initially infected individual by examining the structure of the network [45]. The value is attained by first calculating the probability that a single infection will only lead to

an outbreak of the disease, and not a pandemic. If the infection does not die out (i.e., it is a local outbreak), then it must be a pandemic. That is, we can subtract from 1 the probability of the infection being an outbreak to attain the probability that it will be a pandemic:

$$\Pr[\text{pandemic}] = 1 - \sum_{d=1}^{\infty} p_d (1 + (u-1)T)^d \tag{2}$$

where u is the probability a person interacting with the infected individual does not have the disease. The value for u can be ascertained using traditional root finding methods:

$$u = \frac{\sum_{d=1}^{\infty} d p_d (1 + (u-1)T)^{d-1}}{\sum_{d=1}^{\infty} d p_d} \tag{3}$$

3 Mitigating Disease Spread

3.1 Identifying Critical Individuals

Since constructing a representative contact network is possible, a natural follow-up question is whether one can identify a subset of nodes in the network that are structurally important for the easy transmission of disease throughout the population. In general, this problem is referred to as a problem of detecting critical nodes and a variety of strategies have been proposed that aim to quantify how central or important each node of the network is under different assumptions about the problem context [16, 40, 58]. Here, we investigate the utility of common centrality measures for identifying important individuals to disease spread and subsequently utilize this information to derive potential public policy strategies. These measures do not consider any information about individuals other than their connection structure (number of edges and weights on the edges). Since the network is only a statistical representation of the population, even if highly specialized centrality measures were proposed to actually target individuals for disease mitigation there would be no way to map the solution to reality (unless privacy laws are abolished).

3.2 Network Centrality

We focus on weighted undirected networks $G = (V, E)$ that are simple (contain no multi-edges or self-loops). We assume the network is composed of $|V| = n$ nodes and $|E| = m$ connections. For simplicity in notation we assume that G is defined by its adjacency matrix A:

$$(A_{ij})_{n\times n} = \begin{cases} w_{ij}, & \text{if edge } (i,j) \in E \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where w_{ij} represents the weight of the edge (i.e., contact duration).

A path between vertices $u, v \in G$ is a sequence of edges $\langle u, \ldots, v \rangle$ that one would need to traverse if starting at vertex u and arriving at vertex v. Edges are weighted with the contact duration between adjacent individuals. Thus, the distance between u and v will be the sum of the contact duration along a connecting path. Many paths are possible between pairs of vertices, but the most interesting in this context is the geodesic path, which is the path of shortest total contact duration between $u, v \in V$. Several shortest paths between vertices u, v may exist.

The removal of a vertex $v \in V$ from the graph G is commonly referred to as an attack on the network, but corresponds to a vaccination or quarantine of an individual in the context of disease mitigation. In either case, the remaining graph will be denoted $G(V \setminus \{v\})$, where it is implied that any edge adjacent to v is also removed from the graph when v is attacked. Upon vertex removal the graph may split into a number of connected components whereby no path exists between vertices in different components.

Degree centrality: For vertex i the degree centrality is computed as the sum of the ith row of the adjacency matrix while disregarding edge weights

$$d_i = \sum_{j=1}^{n} \begin{cases} 1, & \text{if } A_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Strength centrality: For vertex i this is determined by summing the weights of all its adjacent edges $s_i = \sum_{j=1}^{n} A_{ij}$.

PageRank: The principal computation behind PageRank [17] is a modification of the basic eigenvector centrality measure by replacing adjacency matrix $\mathbf{A}$ with $\mathbf{A} = \alpha\mathbf{P} + (1-\alpha)\mathbf{Q}$, where matrix $(\mathbf{Q}_{ij})_{n\times n} = 1/n$ captures randomly visiting one of n nodes in the network, and matrix $\mathbf{P}$ models random walks through connections between nodes. The parameter $\alpha \in [0,1]$ is a user-defined dampening parameter usually equal to 0.85. $\mathbf{A}$ will have a unique eigenvector $\mathbf{v} = \mathbf{Av}$ with eigenvalue 1, which is used to rank the nodes by largest v_i value.

Kleinberg's hub and authority scores: When searching for an influential spreader of the disease in a population, one can classify people as either spreaders or those (directly or indirectly) infected by spreaders. The problem is therefore to assign a spreader score and a non-spreader score to each individual. In the lingo of the algorithm, the spreaders are authorities and the non-spreaders are hubs [38].

Calculating the hub and authority scores for each node can be accomplished by an iterative technique. Let $h(i)$ be the hub weight for node i and let $a(i)$ be its authority

weight. The weights are then recursively computed as $h(i) = \sum_{j=1}^{n} A_{ij} a(j) \quad \forall i = 1, \ldots, n$ and $a(j) = \sum_{i=1}^{n} A_{ij} h(i) \quad \forall j = 1, \ldots, n.$

4 Computational Results

In this section we investigate the ability of decision trees to indicate potentially useful public policies. We consider the centrality measures described in Sect. 3.2 for identifying vertices who may be important to the structural integrity of the network. For pandemic disease mitigation the ideal scenario would be one that leads to maximal network destruction by removing the most central vertices, thereby limiting potential avenues for disease to spread and hopefully averting the pandemic entirely. Results are presented based on a single network of the GTA region; however, the results do not significantly differ when averaged over a family of generated networks.

The centrality measure will score each node and we subsequently assign each node a rank based on this score, where lower ranks are considered more important to disease spread. To ensure the decision trees are robustly created a subset of the original data set $\mathcal{D}$ containing the 5.5 million individuals of the GTA is designated as the training set. The training data are determined by selecting the top R centrality ranks, and a new attribute is created for each individual that indicates whether the individual is to be targeted for mitigation or not. That is, each individual will be designated a classification

$$\text{Class}(i) = \begin{cases} 1, & \text{if } rank(i) \leq p \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $rank(i)$ returns the rank of the ith observation in the original data set. Thus, the training set is defined as $\mathcal{Y} = \{x \mid rank(x) \leq p\}$. The full data set $\mathcal{D}$ is employed for testing. In all cases, computing both the centrality measure for the network and an associated decision tree do not require significant time (typically a few seconds if selecting nodes by batch, otherwise a few hours if greedily selecting a node at a time and recomputing the remaining centrality score for the remaining nodes of the network).

4.1 Decision Trees

Decision trees have a number of important advantages over other machine learning methods, including the relatively simple interpretation of resulting classification rules. The robustness of decision trees to combinations of ordinal and numeric data is another important aspect since census information is composed of both types of variables. Moreover, the data set requires minimal preparation before being analyzed and validation is also simple and straightforward through the use of a number of statistical tests. The resulting trees are surprisingly robust to changes in the input data, and for large data sets the computation time needed to construct the trees is minimal.

Another important aspect of decision trees is the ability to easily analyze and adjust the acceptable degree of social disruption. The confusion matrix of a classification tree quantifies the number of correctly and incorrectly classified observations by considering all pairwise possibilities. For binary problems the 2×2 matrix $\mathbf{M}$ has correctly identified observations as diagonal entries (m_{nn} and m_{pp}) and misclassified results are off-diagonal (m_{fp} and m_{fn}):

$$\mathbf{M} = \begin{bmatrix} m_{nn} & m_{fn} \\ m_{fp} & m_{pp} \end{bmatrix} \tag{7}$$

This matrix is particularly useful due to its implications on quickly assessing the potential quality and feasibility of the policies implied by the decision tree. Specifically, false-negative classifications m_{fn} indicate the number of individuals who will not be targeted by the policies, but should have been. The smaller this number is, the more likely the policy adheres to the selection strategy used to identify the nodes. On the other hand, the false-positive m_{fp} entry indicates the number of people who will likely be effected by the policy, but who may not need to have been. These can be interpreted as, or indicative of, the trade-off between population health and socioeconomic costs of the policies.

Incorporating costs into the decision tree model can also be easily accomplished using a cost matrix $\mathcal{L}$, which indicates the cost for misclassifying an individual. Thus, policies can be directly tuned during tree construction by adjusting this cost matrix

$$\mathcal{L} = \begin{bmatrix} 0 & L_{fn} \\ L_{fp} & 0 \end{bmatrix} \tag{8}$$

The choice of cost matrix values is based on whether or not dissemination blocks may be quarantined and whether public transportation routes can be shut down. If both transportation and dissemination blocks are considered then $\mathcal{L}_1$ is used; however, if dissemination block quarantine is possible but transportation route closure is not then $\mathcal{L}_2$ is employed, otherwise $\mathcal{L}_3$ is chosen.

$$\mathcal{L}_1 = \begin{bmatrix} 0 & 9 \\ 10 & 0 \end{bmatrix} \qquad \mathcal{L}_2 = \begin{bmatrix} 0 & 3 \\ 2 & 0 \end{bmatrix} \qquad \mathcal{L}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{9}$$

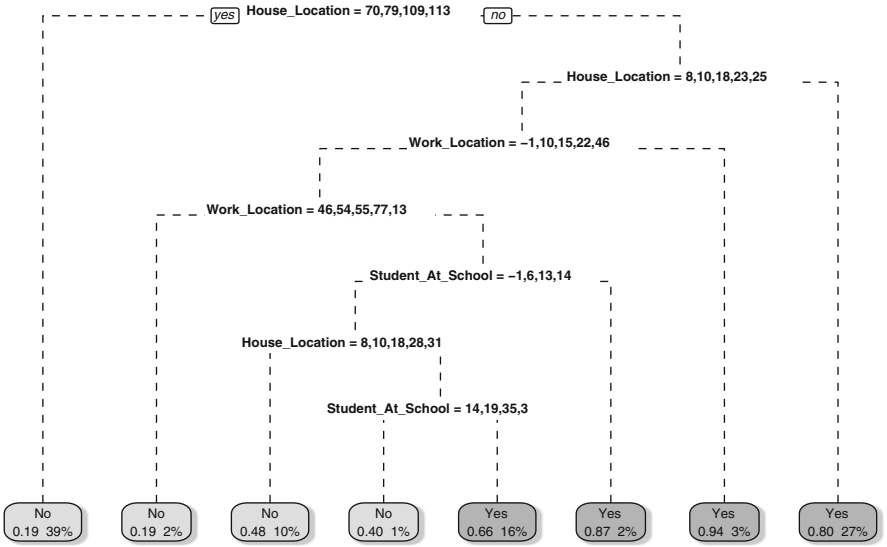
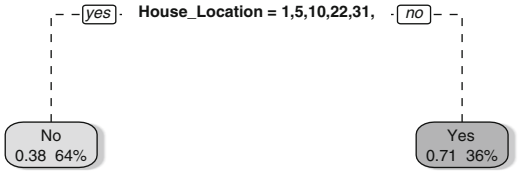


Fig. 3 Decision tree created from individuals identified using degree centrality. Mitigation by dissemination block is permitted, but closing public transport routes is not

Fig. 4 Decision tree created from individuals identified using strength centrality. Mitigation by dissemination block is permitted, as are public transport routes



4.2 Mitigation by Dissemination Block

We examine the consequences of home and work dissemination block information being considering by the decision tree learner. That is, the network centrality measure may identify individuals who live or work in certain dissemination blocks. Figure 3 shows a derived decision tree considering all attributes except those related to public transportation. There are four positive outcomes; however, only two cover a sufficiently large proportion of the population to warrant serious consideration. One rule implies a potential quarantine of homes in certain locations (due to space limitations not all blocks are listed in the diagram) and covers 27 % of the population. The implied rule is able to correctly classify 0.80 of those individuals. The other dominant rule is a combined strategy of quarantine by the household and school dissemination blocks, in addition to targeted school closures. Overall 16 % of the test population was classified according to this rule, which was able to correctly identify 0.66 of those individuals.

A simpler decision tree is derived when additionally considering transportation routes, as shown in Fig. 4. The only split deemed of sufficient relevance was

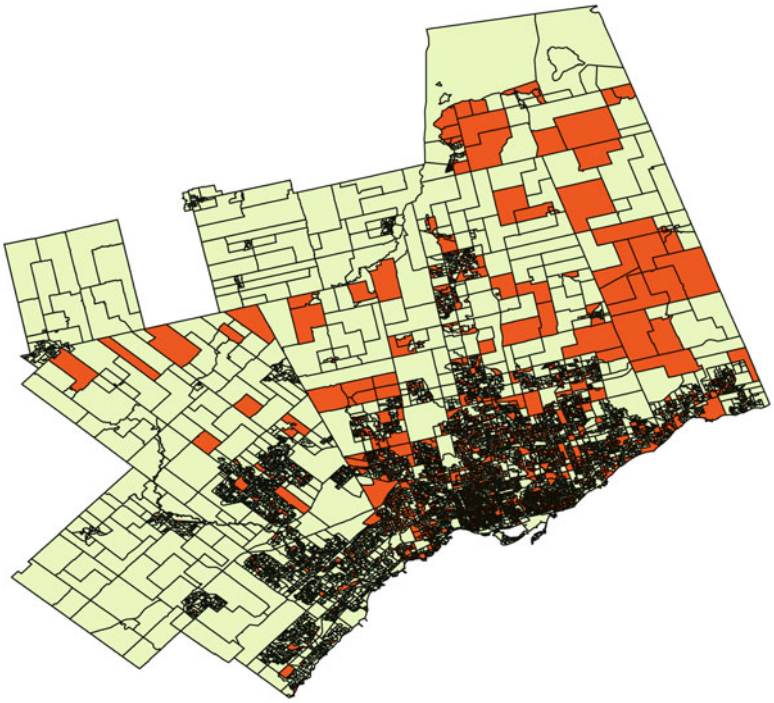


Fig. 5 Map of the GTA indicating all dissemination blocks. Areas of higher population density are smaller and *highlighted blocks* are those identified by the decision tree of Fig. 4

according to house dissemination block. From the test data set only 36 % of individuals are classified using this rule, although 0.71 of them are classified correctly. Figure 5 depicts the GTA and all of the dissemination blocks. The colored areas are those identified by the decision tree shown in Fig. 4 and should be considered for mitigation by some set of public policies. These areas are mostly along major public transit routes. However, the statistically better approach to identifying the individuals was by ignoring the transportation route attribute.

4.3 Not Considering Dissemination Block

Targeting by dissemination block may be an effective idea in theory but is impractical due to the socioeconomic disturbance such policies could create. We therefore also construct decision trees that do not include the dissemination block information as a feature of each individual. This will force the utilization of other attributes such as school location and age. School closures are a common consideration when pandemic planning. However, through this approach a more fine-tuned selection of schools may be provided to policy makers.

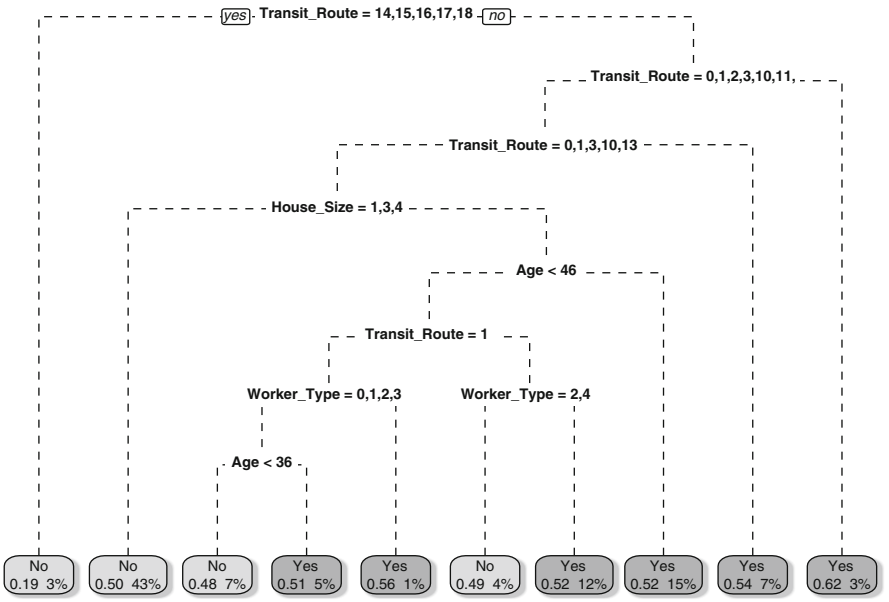


Fig. 6 Decision tree created from individuals identified using strength centrality. Mitigation by dissemination block is not permitted, but public transportation routes may be closed

Figure 6 presents an example decision tree created based on strength centrality and also suggests the closure of some public transportation routes. All of the discovered rules involve closing a subset of routes, but targeting by age group, type of job, and number of people living in a household is also deemed useful targeting information. Other centrality measures also led to more specific trees in these data context and included rules focusing on hospitals and distance traveled, for example.

The decision tree shown in Fig. 7 was created based on degree centrality and does not allow the closure of public transportation routes. The tree has 13 rules; however, of those with positive outcomes only two cover 9 % or greater of the test population. Both rules focus on school closures. The first is only targeted to schools and has a correct classification rate of 0.77. The other rule also closes a subset of schools, but additionally targets working-age individuals who live within a 44,000 m radius from their workplace dissemination block. Figure 8 highlights the exact schools that were targeted for closure by the first rule.

4.4 Effect on the Potential Pandemic

We provide an analysis of the residual network after a policy has been put in place. That is, we consider decision tree branches that lead to a positive outcome and that also cover at least 9 % of the training data at a correct classification rate

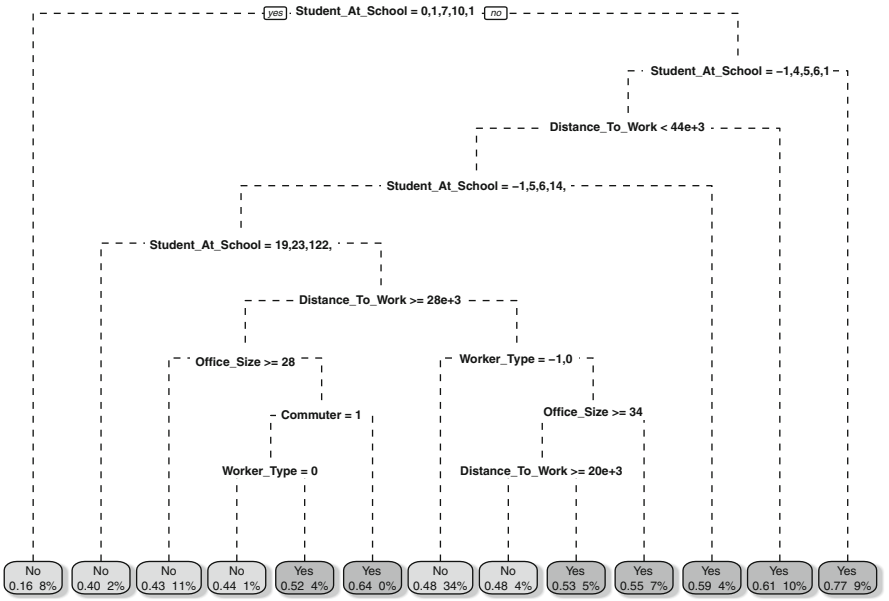


Fig. 7 Decision tree created from individuals identified using degree centrality. Mitigation by dissemination block is permitted, but public transportation routes are not

of at least 0.5. These rules are then forced upon the graph topology, resulting in the removal of a number of network connections. The rules were chosen from decision trees that use 20 % of the training during the learning stage. We do not consider policies that can be targeted towards dissemination blocks. Table 2 presents some example rules that are implied from each decision tree. Interestingly, strength centrality only selected transit routes as targets, whereas each of the three other centrality attack strategies suggested a mixture of individuals' attributes.

Table 3 shows the effect of each rule given in Table 2 with respect to properties of the residual network after implementing the policy, as well as $\mathcal{R}_0$ and the probability of a pandemic occurring. For these results we set $T = 0.1$, which corresponds to a disease that spreads with relative but not extreme ease. For each strategy we also examine the impact of individual compliance on the ability of the implied policy to mitigate disease spread. We distinguish between policies that can be complied to and those that are imposed, for example, targeting individuals who travel a certain distance to work versus targeting transportation routes. In the latter case, the implication is that the route is closed whereas the former is impossible to restrict in reality and is left to the individual as to whether to comply with such a public policy. Any member for whom the policy applies will comply with it with a certain probability, which we vary over $[0, 1]$. Each strategy is able to significantly decrease the basic reproduction number $\mathcal{R}_0$ and the probability of a pandemic occurring to approximately 0.5. Pagerank and degree centrality have the additional benefits of decreasing the largest graph component to approximately half of the original 5.5

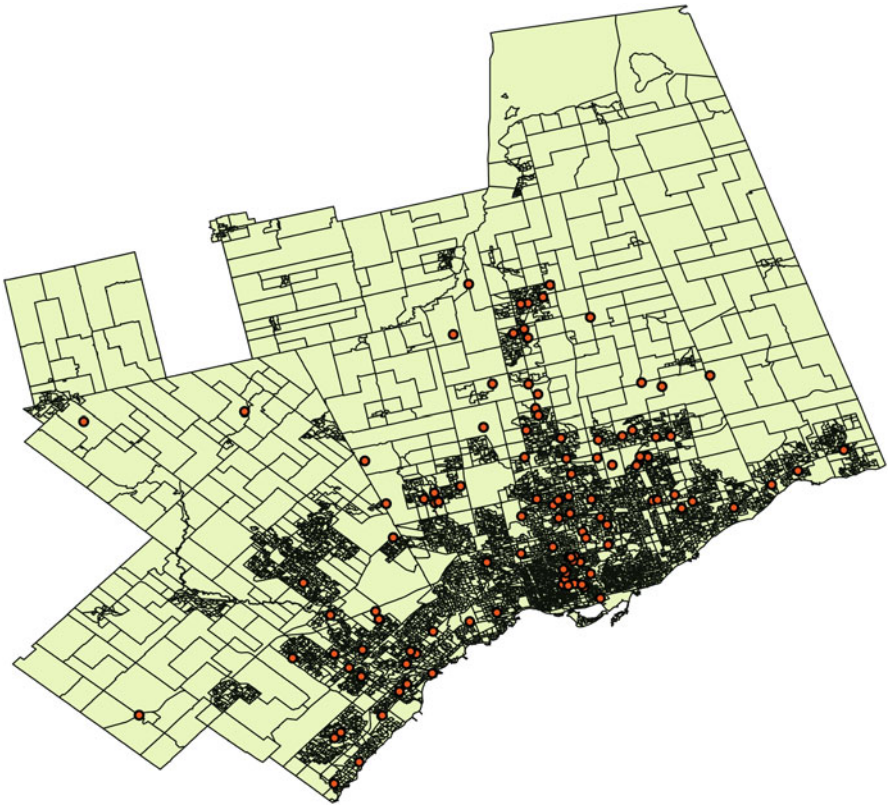


Fig. 8 The GTA with school closures *highlighted* in accordance with one of the implied rules in Fig. 7

million individuals. Thus, each of these strategies may be reasonable candidates as a basis for actual policies. The overall properties of PageRank indicate that if 100 % compliance is reached then it will have generated the most useful strategies among the centrality measures considered, with degree centrality and authority close afterwards. However, authority centrality yields more desirable results for lower amounts of compliance. Figure 9 presents the decrease in probability of a pandemic occurring with respect to the probability of public compliance to the policies.

5 Discussion and Conclusions

Pandemic diseases have the potential to devastate large populations by causing enormous socioeconomic costs and an unimaginable number of deaths. Unfortunately, we do not yet understand well enough the dynamics and evolution of pathogens

Table 2 Example policy guidelines implied by the decision tree for each of the four centrality measures considered

Strategy	Policy guidelines
Authority	Transit route $\in \{0,1,2,3,6,8,10,12,13\}$
	Worker type $\in \{\text{NA}, \text{Office/Clerical}, \text{High volume}, \text{Low volume}\}$
	Age $\in [25,45]$
	Clinician Type $\in \{\text{Doctor}\}$
	Hospital $\in \{0,\ldots,9\}$
	Office size < 417
	House size $\in [3,6^{+}]$
Coverage: 31 %, classification rate: 0.50	
Degree	Transit route $\in \{0,1,2,3,10,11,12,13\}$
	Distance to work $< 24{,}500$ m
	House size $>= 3$
	Student at School (set too large to list)
	Office size $\in [7,287]$
	Age ≥ 41
Coverage: 18 %, classification rate: 0.53	
PageRank	Transit route $\in \{1,3,11,12,13\}$
	Teacher at school of size $< 2,166$
	Distance to work $\in [6650,17500]$ m
	Age > 32
	Office size $\in [6,491]$
	Hospital $\in \{0,1,2,4,5,6,10\}$
Coverage: 9 %, classification rate: 0.55	
Strength	Transit route $\in \{0,1,2,3,4,5,6,7,8,9,10,11,12,13\}$
Coverage: 97 %, classification rate: 0.51	

to be able to predict when or where the next pandemic will begin, or even how it will transmit. Consequently, a range of efforts have focused on creating efficient pandemic mitigation plans and tools for use by public health officials so that they may be better prepared when pandemics are detected. We presented a novel approach that combines the use of centrality measures as a proof of concept that targeting individuals based on residual social contact network topology may be a viable method for determining appropriate subsets of individuals to target policies towards in order to contain disease spread. Subsequent data mining of census data using decision trees is performed on the subset of identified individuals in order to provide potentially informative feedback to policy decision makers. It must be stressed that the presented work is proof-of-concept and a number of advancements and practical considerations must be incorporated before the tool can provide truly insightful information.

Table 3 Summary statistics of residual networks after policies of Table 2 have been implemented, and individuals comply with the policy according to a certain rate

Strategy	Comp	# Edges	μ_{degree}	σ_{degree}	max$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$\mathcal{R}_0$	PrP
Authority	0.0	47,225,721	17.25	29.78	5,476,158	1	2.404	0.68
Authority	0.1	44,529,165	16.28	29.93	5,470,659	103	2.447	0.64
Authority	0.2	43,150,278	15.81	30.30	5,458,893	648	2.483	0.62
Authority	0.3	42,259,472	15.53	30.97	5,438,402	1,795	2.523	0.60
Authority	0.4	41,598,725	15.37	31.87	5,406,441	3,805	2.587	0.58
Authority	0.5	41,062,673	15.29	33.02	5,355,687	7,083	2.647	0.57
Authority	0.6	40,629,770	15.32	34.43	5,276,420	12,365	2.716	0.55
Authority	0.7	40,272,317	15.51	36.17	5,147,377	20,953	2.802	0.54
Authority	0.8	39,977,696	15.94	38.30	4,936,971	34,719	2.901	0.54
Authority	0.9	39,731,849	16.81	41.04	4,634,278	42,867	3.005	0.55
Authority	1.0	39,543,434	18.55	44.88	4,262,366	47,264	2.912	0.53
Degree	0.0	100,351,408	36.65	34.47	5,476,158	1	4.811	0.90
Degree	0.1	90,049,222	32.89	32.95	5,475,995	1	4.483	0.87
Degree	0.2	82,062,670	29.98	31.54	5,475,276	4	4.222	0.84
Degree	0.3	74,791,671	27.33	30.23	5,472,936	12	3.984	0.81
Degree	0.4	67,886,974	24.84	29.04	5,466,592	76	3.772	0.78
Degree	0.5	61,227,165	22.46	27.82	5,450,478	388	3.577	0.74
Degree	0.6	54,783,296	20.24	26.76	5,410,985	1,380	3.404	0.70
Degree	0.7	48,443,959	18.20	25.88	5,311,878	5,173	3.237	0.65
Degree	0.8	42,237,379	16.56	25.31	5,063,532	17,538	3.079	0.61
Degree	0.9	36,236,150	15.85	25.46	4,478,214	43,629	2.910	0.59
Degree	1.0	30,608,045	19.28	28.05	3,175,502	49,836	2.711	0.55
PageRank	0.0	101,424,113	37.04	42.44	5,476,155	1	4.511	0.90
PageRank	0.1	88,934,391	32.48	39.60	5,475,837	1	4.138	0.86
PageRank	0.2	79,206,265	28.93	36.91	5,474,869	6	3.854	0.82
PageRank	0.3	70,638,981	25.82	34.37	5,471,872	29	3.651	0.78
PageRank	0.4	62,579,887	22.91	32.00	5,463,963	125	3.405	0.74
PageRank	0.5	54,882,960	20.16	29.68	5,442,846	483	3.213	0.68
PageRank	0.6	47,474,574	17.62	27.62	5,385,537	2,014	3.054	0.62
PageRank	0.7	40,234,487	15.36	25.95	5,221,545	8,313	2.907	0.57
PageRank	0.8	33,392,418	13.75	24.98	4,782,891	33,309	2.728	0.52
PageRank	0.9	27,124,492	13.49	25.61	3,835,836	84,381	2.558	0.52
PageRank	1.0	21,530,549	17.99	30.49	2,393,549	96,531	2.407	0.47
Strength	0.0	95,540,686	34.89	42.34	5,476,155	1	4.163	0.90

The compliance probability *Comp*, mean degree μ_{degree}, standard deviation of degree σ_{degree}, largest component max $\mathcal{C}$, number of components $\lvert\mathcal{C}\rvert$, reproductive number $\mathcal{R}_0$, and probability of a pandemic PrP are all reported

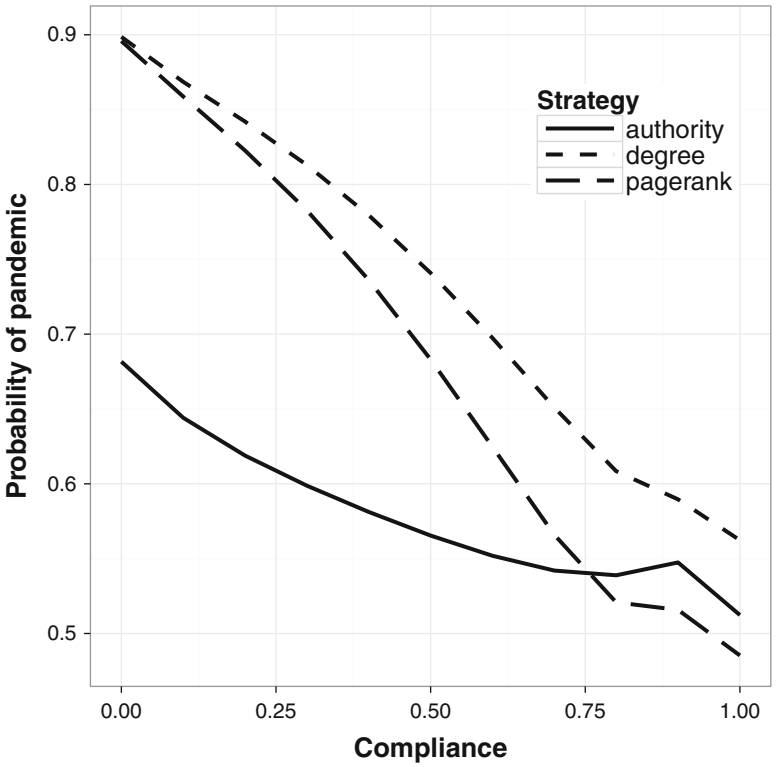


Fig. 9 Probability of pandemic occurring as a function of compliance to the public policies in Table 2

Incorporating real-world cost estimates into the model could allow for better tuning of the cost matrix used to determine decision tree splits. Additionally, more information about the costs of real-world scenarios (such as advertising campaigns) may provide enough information to begin measuring the potential implementation strategies that public policy makers can take in order to target the indicated individuals. The longer term goal would be for the system to return a portfolio of immediately implementable strategies to the decision maker, with already performed cost-benefit analyses, etc.

The centrality measures used in this study to target individuals were chosen because they have been shown to work well in other domains for similar network attack problems. We find that they may be useful in the pandemic context as well, but considering more advanced formulations of the problem is likely to result in improved performance. Another possible consideration is timing. Specifically, determining when policies should be implemented and what observations of public behavior are important to trigger certain policies in order to achieve the optimal balance between minimizing socioeconomic impact and maximizing disease mitigation.

References

1. Ajelli, M., Merler, S.: The impact of the unstructured contacts component in influenza pandemic modeling. PLoS One **3**(1), e1519 (2008)
2. Aleman, D.M., Wibisono, T.G., Schwartz, B.: A nonhomogeneous agent-based simulation approach to modeling the spread of disease in a pandemic outbreak. Interfaces **41**(3), 301–315 (2011)
3. Ames, G.M., George, D.B., Hampson, C.P., Kanarekand, A.R., McBee, C.D., Lockwood, D.R., Achter, J.D., Webb, C.T.: Using network properties to predict disease dynamics on human contact networks. Proc. R. Soc. Biol. **278**(1724), 3544–3550 (2011)
4. Anderson, R.M., May, R.M.: Infectious Diseases of Humans: Dynamics and Control. Oxford University Press, Oxford (1992)
5. Aparicio, J.P., Pascual, M.: Building epidemiological models from r0: an implicit treatment of transmission in networks. Proc. R. Soc. B **274**(1609), 505–512 (2007)
6. Aurenhammer, F.: Voronoi diagrams a survey of a fundamental geometric data structure. ACM Comput. Surv. **23**(3), 345–405 (1991)
7. Bansal, S., Pourbohloul, B., Meyers, L.A.: A comparative analysis of influenza vaccination programs. PLoS Med. **3**(10), e387 (2006)
8. Bansal, S., Grenfell, B.T., Meyers, L.A.: When individual behaviour matters: homogeneous and network models in epidemiology. J. R. Soc. Interface **4**(16), 879–891 (2007)
9. Barrett, C.L., Bisset, K.R., Eubank, S.G., Feng, X., Marathe, M.V.: Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, pp. 1–12 (2008)
10. Bartlett , J.G., Borio, L.: The current status of planning for pandemic influenza and implications for health care planning in the united states. Clin. Infect. Dis. **46**(6), 919–925 (2008)
11. Basta, N.E., Chao, D.L., Halloran, M.E., Matrajt, L., Longini, I.M.: Strategies for pandemic and seasonal influenza vaccination of schoolchildren in the United States. Am. J. Epidemiol. **170**(6), 679–686 (2009)
12. Becker, N.G., Hasofer, A.M.: Estimating the transmission rate for a highly infectious disease. Biometrics **54**(2), 730–738 (1998)
13. Bettencourt, L.M.A., Ribeiro, R.M.: Real time bayesian estimation of the epidemic potential of emerging infectious diseases. PLoS One **3**(5), e2185 (2008)
14. Blake, K.D., Blendon, R.J., Viswanath, K.: Employment and compliance with pandemic influenza mitigation recommendations. Emerg. Infect. Dis. **16**(2), 212–218 (2010)
15. Blendon, R.J., Koonin, L.M., Benson, J.M., Cetron, M.S., Pollard, W.E., Mitchell, E.W., Weldon, K.J., Herrmann, M.J.: Rescinding community mitigation strategies in an influenza pandemic. Emerg. Infect. Dis. **14**(5), 778–786 (2008)
16. Borgatti, S.P., Everett, M.G.: A graph-theoretic perspective on centrality. Soc. Netw. **28**(4), 466–484 (2006)
17. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**, 107–117 (1998)
18. Chao, D.L., Halloran, M.E., Longini, I.M., Jr.: School opening dates predict pandemic influenza a(h1n1) outbreaks in the united states. J. Infect. Dis. **202**(6), 877–880 (2010)
19. Chowell, G., Viboud, C., Wang, X., Bertozzi, S.M., Miller, M.A.: Adaptive vaccination strategies to mitigate pandemic influenza: Mexico as a case study. PLoS One **4**(12), e8164 (2009)
20. Cintron-Arias, A., Castillo-Chavez, C., Bettencourt, L.M.A., Lloyd, A.L., Banks, H.T.: The estimation of the effective reproductive number from disease outbreak data. Math. Biosci. Eng. **6**(2), 261–282 (2009)
21. Coelho, F.C., Cruz, O.G., Codeco, C.T.: Epigrass: a tool to study disease spread in complex networks. Source Code Biol. Med. **3**(3) (2008) doi: 10.1186/1751-0473-3-3

22. Cori, A., Boelle, P.-Y., Thomas, G., Leung, G.M., Valleron, A.-J.: Temporal variability and social heterogeneity in disease transmission: the case of sars in Hong Kong. PLoS Comput. Biol. **5**(8), e1000471 (2009)
23. Daems, R., Del Giudice, G., Rappuoli, R.: Anticipating crisis: towards a pandemic flu vaccination strategy through alignment of public health and industrial policy. Vaccine **23**, 5732–5742 (2005)
24. Danon, L., Ford, A.P., House, T., Jewell, C.P., Keeling, M.J., Roberts, G.O., Ross, J.V., Vernon, M.C.: Networks and the epidemiology of infectious disease. Int. Perspect. Infect. Dis. **2011**, 284909 (2011)
25. Del Valle, S.Y., Stroud, S.Y., Smith, J.P., Mniszewski, S.M., Riese, J.M., Sydoriak, S.J.: Episims: epidemic simulation system. Technical report, Los Almos National Laboratory, NM (2006)
26. Del Valle, S.Y., Hyman, J.M., Hethcote, H.W., Eubank, S.G.: Mixing patterns between age groups in social networks. Soc. Netw. **29**, 539–554 (2007)
27. Eubank, S., Gucle, H., Kumar, V.S.A., Marathe, M.V., Srinivasan, A., Toroczkal, Z., Wang, N.: Modelling disease outbreaks in realistic urban social networks. Nature **429**, 180–184 (2004)
28. Eubank, S., Anil Kumar, V.S., Marathe, M., Srinivasan, A., Wang, N.: Structure of social contact networks and their impact on epidemics. In: AMS-DIMACS Special Issue on Epidemiology, pp. 181–213 (2006)
29. Eubank, S., Barrett, C., Beckman, R., Bisset, K., Durbeck, L., Kuhlman, C., Lewis, B., Marathe, A., Marathe, M., Stretz, P.: Detail in network models of epidemiology: are we there yet? J. Biol. Dyn. **4**(5), 446–455 (2010)
30. Farrington, C.P., Kanaan, M.N., Gay, N.J.: Estimation of the basic reproduction number for infectious diseases from age-stratified serological survey data. J. R. Stat. Soc. Ser. C Appl. Stat. **50**(3), 251–292 (2001)
31. Ferguson, N.M., Cummings, D.A.T., Fraser, C., Cajka, J.C., Cooley, P.C., Burke, D.S.: Strategies for mitigating an influenza pandemic. Nature **442**, 448–452 (2006)
32. Ferrari, M.J., Bjornstad, O.N., Dobson, A.P.: Estimation and inference of {R0} of an infectious pathogen by a removal method. Math. Biosci. **198**(1), 14–26 (2005)
33. Fumanelli, L., Ajelli, M., Manfredi, P., Vespignani, A., Merler, S.: Inferring the structure of social contacts from demographic data in the analysis of infectious diseases spread. PLoS Comput. Biol. **8**(9), e1002673 (2012)
34. Glass, L., Glass, R.: Social contact networks for the spread of pandemic influenza in children and teenagers. BMC Publ. Health **8**(1), 61 (2008)
35. Haber, M.J., Shay, D.K., Davis, X.M., Patel, R., Jin, X., Weintraub, E., Orenstein, E., Thompson, W.W.: Effectiveness of interventions to reduce contact rates during a simulated influenza pandemic. Emerg. Infect. Dis. **13**(4), 581–589 (2007)
36. Kelso, J., Halder, N., Milne, G.: Vaccination strategies for future influenza pandemics: a severity-based cost effectiveness analysis. BMC Infect. Dis. **13**(1), 81 (2013)
37. Kermak, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics. Proc. R. Soc. Lond. **115**, 700–721 (1927)
38. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
39. Kok, G., Jonkers, R., Gelissen, R., Meertens, R., Schaalma, H., de Zwart, O.: Behavioural intentions in response to an influenza pandemic. BMC Publ. Health **10**(1), 174 (2010)
40. Landherr, A., Friedl, B., Heidemann, J.: A critical review of centrality measures in social networks. Bus. Inf. Syst. Eng. **2**(6), 371–385 (2010)
41. Li, J., Blakeley, D., Smith, R.J.: The failure of r0. Comput. Math. Methods Med. **527610** (2011)
42. Longini, I.M., Jr., Nizam, A., Xu, S., Ungchusak, K., Hanshaoworakul, W., Cummings, D.A.T., Halloran, M.E.: Containing pandemic influenza at the source. Science **309**, 1083–1088 (2005)
43. Milne, G., Kelso, J., Kelly, H.: Strategies for mitigating an influenza pandemic with pre-pandemic h5n1 vaccines. J. R. Soc. Interface **7**, 573–586 (2010)

44. Mossong, J., Hens, N., Jit, M., Beutels, P., Auranen, K., Mikolajczyk, R., Massari, M., Salmaso, S., Tomba, G.S., Wallinga, J., Heijne, J., Sadkowska-Todys, M., Rosinska, M., Edmunds, W.J.: Social contacts and mixing patterns relevant to the spread of infectious diseases. PLoS Med. **5**(3), e74 (2008)
45. Newman, M.E.J.: Spread of epidemic disease on networks. Phys. Rev. E **66**, 016128 (2002)
46. Oshitani, H.: Potential benefits and limitations of various strategies to mitigate the impact of an influenza pandemic. J. Infect. Chemother. **12**(4), 167–171 (2006) doi: 10.1155/2011/527610
47. Osterholm, M.T.: Preparing for the next pandemic. N. Engl. J. Med. **352**(18), 1839–1842 (2005)
48. Potter, C.W.: Chronicle of influenza pandemics. In: Nicholson, K.G., Webster, R.G., Hay, A.J. (eds.) Textbook of Influenza, pp. 3–18. Blackwell Science, Oxford (1998)
49. Roy, M., Pascual, M.: On representing network heterogeneities in the incidence rate of simple epidemic models. Ecol. Complex. **3**(1), 80–90 (2006)
50. Salathe, M., Kazandjieva, M., Lee, J.W., Levis, P., Feldman, M.W., Jones, J.H.: A high-resolution human contact network for infectious disease transmission. Proc. Natl. Acad. Sci. **107**(51), 22020–22025 (2010)
51. Schwartz, B., Gellin, B.: Vaccination strategies for an influenza pandemic. J. Infect. Dis. **191**(8), 1207–1209 (2005)
52. Spear, R.C., Hubbard, A.: Parameter estimation and site-specific calibration of disease transmission models. In: Michael, E., Spear, R.C., (eds.) Modelling Parasite Transmission and Control. Advances in Experimental Medicine and Biology, vol. 673, pp. 99–111. Springer, New York (2010)
53. StatsCan. Canadian census 2006. Technical report, Statistics Canada (2007)
54. Tatem, A.J., Rogers, D.J., Hay, S.I.: Global transport networks and infectious disease spread. Adv. Parasitol. **62**, 293–343 (2006)
55. Toronto Transit Commission. Usage statistics 2011–2012. http://www.ttc.ca/ (2013)
56. Tsai, M., Chern, T., Chuang, J., Hsueh, C., Kuo, H., Liau, C., Riley, S., Shen, B., Shen, C., Wang, D., Hsu, T.: Efficient simulation of the spatial transmission dynamics of influenza. PLoS One **5**(11), e13292 (2010)
57. Tuite, A., Fisman, D.N., Kwong, J.C., Greer, A.: Optimal pandemic influenza vaccine allocation strategies for the Canadian population. PLoS One **5**(5), PMC2804393 (2010)
58. Valente, T.W., Coronges, K., Lakon, C., Costenbader, E.: How correlated are network centrality measures? Connections (Toronto, ON) **28**(1), 16–26 (2008)
59. Ventresca, M., Aleman, D.: Evaluation of strategies to mitigate contagion spread using social network characteristics. Soc. Netw. **35**(1), 75–88 (2013)
60. Wallinga, J., Teunis, P., Kretzschmar, M.: Using data on social contacts to estimate age-specific transmission parameters for respiratory-spread infectious agents. Am. J. Epidemiol. **164**(10), 936–944 (2006)
61. Wallinga, J., van Boven, M., Lipsitch, M.: Optimizing infectious disease interventions during an emerging epidemic. Proc. Natl. Acad. Sci. **107**(2), 923–928 (2010)
62. Wu, J.T., Leung, G.M., Lipsitch, M., Cooper, B.S., Riley, S.: Hedging against antiviral resistance during the next influenza pandemic using small stockpiles of an alternative chemotherapy. PLoS Med. **6**(5), e1000085 (2009)

On an Asymptotic Property of a Simplicial Statistical Model of Global Optimization

Antanas Žilinskas and Gražina Gimbutienė

Abstract A homogeneous isotropic Gaussian random field is accepted as a statistical model of objective functions, aiming to construct global optimization algorithms. The asymptotic of the conditional mean and variance is considered, assuming that the random field values are known at the vertices of a simplex, and that the latter is contracting. The obtained result theoretically substantiates the construction of the recently proposed bi-variate global optimization algorithm, which arouses interest due to good performance in testing experiments and the established convergence rate. The obtained result also enhances motivation to extend the aforementioned algorithm to higher dimensions.

Keywords Statistical models for global optimization • Black-box optimization • Simplicial statistical models

1 Introduction

A statistical model for global optimization is considered, where the objective function is available as a black box, i.e. the analytic expression of the objective function is not known and only the computation of function values is allowable. Further, it is assumed that a lot of time is needed to compute a single function value. In such a situation a conceptual design of an optimization algorithm is difficult. We consider the development of an algorithm based on the rational decision theory and statistical models of uncertainty. More precisely, an asymptotic property of the statistical model is proved which theoretically substantiates a computational simplification of the recently proposed methods [2, 20].

For the basics of statistical model-based approach to global optimization, we refer to the following monographs [9, 13, 14, 16]. The main advantages of the proposed algorithms are their optimality with respect to such criteria as the

A. Žilinskas (✉) • G. Gimbutienė
Institute of Mathematics and Informatics, Vilnius University, Akademijos 4, 08663 Vilnius, Lithuania
e-mail: antanas.zilinskas@mii.vu.lt; grazina.gimbutiene@mii.vu.lt

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_20

maximum mean improvement, maximum improvement probability, and maximum information. However, the applicability of the original versions of these algorithms is limited to the optimization of surely expensive objective functions. The latter disadvantage is caused by the inherent complexity of the algorithms implying their computationally intensive implementations. Therefore, simplifications of auxiliary computational problems here are especially valuable. Similarly, original versions of the Lipschitzian algorithms also involve intensive auxiliary computations [5]. However, by implementing various simplifications advanced Lipschitzian algorithms have been developed that are efficient for rather a broad field of applications with not necessarily expensive objective functions. For example, from the point of view of implementation simplicity, the diagonal algorithms [10, 11] are especially attractive. The implementation advantages can also be gained by selecting an appropriate partition method [1, 3, 6–8, 12, 15]. Similar ideas can also be helpful in reducing the computational complexity of statistical model-based global optimization algorithms. In the present paper, we focus on the statistical models of global optimization related to simplicial partition of the feasible region.

The simplicial version of statistical models is promising in the construction of global optimization algorithms, as shown, e.g., in [21]. A subsequent heuristic simplification of computations, proposed in [2], considerably reduces the computational burden. In the present paper, we provide a theoretical foundation and some generalisation of that simplification.

2 Motivation of the Research

The global minimisation problem $\min_{X\in\mathbf{A}} f(x)$, $\mathbf{A}\subset\mathbb{R}^d$, is considered, where $f(x)$ is a continuous function and A is a compact set. It is assumed that $f(x)$ is available either as a complicated computational model or unfamiliar software. Consequently, the analytic properties of $f(x)$ are unavailable and unfavourable properties of $f(x)$ such as non-differentiability, non-convexity or multi-modality cannot be ruled out. In such a situation, to construct an optimization algorithm in the frame of the theory of rational decisions under uncertainty [4], a model of uncertainty and a utility function of an optimizer are needed. The corresponding axiomatic, proposed in [17, 18], validates the use of a family of random variables ξ_x, $x\in\mathbf{A}$, as a statistical model of the objective function and the so-called P-algorithm. The latter is defined at the n-th minimisation step as follows. The values of the objective function $y_i = f(x_i)$ are supposed to be computed at the previous optimization steps, where $x_i\in\mathbf{A}$, $i=1,\ldots,n$, and an unknown value $f(x)$, $x\neq x_i$, is interpreted as a random variable ξ_x. The current function value is computed at the point of maximum probability to improve the solution found at the previous optimization steps:

$$x_{n+1} = \arg\max_{x\in\mathbf{A}} \mathbf{P}\{\xi(x)\le y_{on}\,|\,\xi(x_1)=y_1,\ldots,\xi(x_n)=y_n\}, \tag{1}$$

where the probability in question is defined by the chosen statistical model and $y_{on} < \min\limits_{1\le i\le n} y_i$ is a minimal known objective function value desirable to get at the $(n+1)$-th minimisation step.

Normally the random variables ξ_x are assumed to be distributed according to the Gaussian probability density. In that case the probability in (1) is defined by the formula

$$\mathbf{P}\{\xi(x) \le y_{on} \,|\, \xi(x_i) = y_i, i = 1,\dots,n\} = \Phi\left(\frac{y_{on} - m(x|(x_i,y_i), i = 1,\dots,n)}{s(x|(x_i,y_i), i = 1,\dots,n)}\right), \tag{2}$$

where $\Phi(\cdot)$ denotes the Gaussian cumulative distribution function, while $m(\cdot)$ and $s^2(\cdot)$ denote the conditional mean and conditional variance of ξ_x, respectively.

Since $\Phi(t)$ is monotonically increasing, the maximum in (1) can be found as follows:

$$x_{n+1} = \arg\max_{x\in\mathbf{A}} \left(\frac{y_{on} - m(x|(x_i,y_i), i = 1,\dots,n)}{s(x|(x_i,y_i), i = 1,\dots,n)}\right). \tag{3}$$

The recent results in [2, 20] validate the algorithm (3) by simpler arguments than those in [17, 18] and without the assumption of Gaussian distribution. Nevertheless, the computations of $m(\cdot)$ and $s(\cdot)$ are relatively complicated, implying time-consuming auxiliary computations of the considered algorithm.

Simplicial statistical models have been invented to simplify the implementation of the corresponding algorithms. The two-dimensional P-algorithm, based on a simplicial statistical model, was originally proposed in [19] under the title "select and clone". It was implemented as a sequential triangular partition of the feasible region. The optimization starts from covering the feasible region by equilateral triangles and computing the objective function values at their vertices. Further, the algorithm loops over triangles, selecting one of them for subdivision. The selection criterion is similar to that used by the original P-algorithm to find the point for the current computation of the objective function value (3). However, the conditional mean and variance are computed with respect to the known function values at the vertices of the triangle. The maximisation over the complete feasible region at every optimization step (2) is replaced by the maintenance of the priority queue comprised of triangle sub-regions of $\mathbf{A}$, where the priority of the triangle S_j is related to the maximum improvement probability

$$p_j = \frac{y_{on} - m(x_{cj}|(x_i,y_i), x_i \in S_j)}{s(x_{cj}|(x_i,y_i), x_i \in S_j)}, \tag{4}$$

where x_{cj} is the weight centre of S_j. For the generalisation of this algorithm as $d > 2$, we refer to [16].

A further reduction of computational burden proposed in [2] for two-dimensional $(d = 2)$ algorithms is also based on triangular partitions of the feasible region.

However, the proposed algorithm differs from the "select and clone" by the partition method and by the priority criterion. The Delaunay triangulation is used for partition of the feasible region. The priority criterion of $S_j \in \mathbf{A}$ is defined by the following formula

$$\pi_j = \frac{Q_j}{m_j - y_{on}}, \tag{5}$$

where m_j is the average of objective function values computed at the vertices of S_j, and Q_j is the area of S_j.

The high convergence rate of that algorithm has been shown in [2] as well as its good performance in the testing experiments. Since the priority criterion (5) was justified in [2] only heuristically, it was important to continue the investigation aimed at the mathematical proving the relation between (5) and the improvement probability (2). Thus, we want to show that, under some assumptions, the application of criteria (4) and (5) is equivalent. In the follow-up sections, the relevant mathematical results are presented.

3 Statement of the Problem

Let $\xi(x)$, $x \in \mathbf{A} \subseteq \mathbb{R}^d$, be a homogeneous isotropic Gaussian random field with the mean value μ, variance σ^2, and the correlation function $\rho(t) = \exp(-ct^2)$. At the points $a_j \in \mathbf{A}$ the values of $\xi(x)$ are known: $\xi(a_j) = z_j$, where $a_j, j = 1, \ldots, n+1$, are vertices of a regular n-simplex with the edge length equal to δ. Let a be the weight centre of the simplex. We are interested in the behaviour of the conditional mean and conditional variance of $\xi(a)$ when the edge length δ of the simplex vanishes. To compute the conditional mean and variance, the correlation coefficients between $\xi(a_i)$ and $\xi(a_k)$, as well as between $\xi(a_i)$ and $\xi(a)$, are needed. The former is obviously equal to $\rho_{ik} = \exp(-c\delta^2)$, and the latter is equal to $\rho_i = \exp(-c\delta^2 \frac{d}{2(d+1)})$ (since the distance between a and a_i is equal to $\delta\sqrt{\frac{d}{2(d+1)}}$). In the upcoming discussion, the following notation and formulas of conditional mean and conditional variance will be used:

$$E(\xi(a)|\xi(a_j) = z_j, j = 1, \ldots, d+1) = m(a|(a_j, z_j), j = 1, \ldots, d+1) =$$
$$= M(\delta, Z) = \mu + \exp\left(-c\delta^2 \frac{d}{2(d+1)}\right) \cdot I^T \cdot C^{-1} \cdot (Z - \mu I), \tag{6}$$

$$Var(\xi(a)|\xi(a_j) = z_j, j = 1, \ldots, d+1) = s^2(a|(a_j, z_j), j = 1, \ldots, d+1) =$$
$$= S^2(\delta) = \sigma^2 \left(1 - \exp\left(-c\delta^2 \frac{d}{d+1}\right) \cdot I^T \cdot C^{-1} \cdot I\right), \tag{7}$$

where I denotes the $(d+1)$-dimensional unit vector, $Z = (z_1, \ldots, z_{d+1})^T$, and C is a $(d+1) \times (d+1)$ matrix with all the elements equal to $\exp(-c\delta^2)$ except the diagonal elements which are equal to 1.

The improvement probability related criterion (4) for the considered simplex can be expressed by the following formula

$$p(\delta) = \frac{y_{on} - M(\delta, Z)}{S(\delta)}, \tag{8}$$

where the essential variables are presented explicitly. To avoid computations with fractions, the denominators of which are close to zero, instead of $p(\delta)$ its reciprocal value with an inverse sign

$$\tilde{p}(\delta) = \frac{S(\delta)}{M(\delta, Z) - y_{on}}, \tag{9}$$

is used for the implementation of the algorithm, as well as in a further analysis.

Formula (5), adapted to the special case considered, can be written as follows:

$$\pi(\delta) = \frac{Q}{\tilde{z} - y_{on}}, \tag{10}$$

where $\tilde{z}$ denotes the average of z_i, $i = 1, \ldots, d+1$, Q is the area of the considered simplex and $y_{on} < \min\limits_{1 \le i \le d+1} z_i$.

We aim to show that (10) well approximates the special case ($d = 2$) of (9) for small δ. It is expected that, for small δ, (9) can be approximated by an expression similar to (5) in the case $d > 2$ as well.

4 Assessment of Approximation

It is obvious that both criteria (5) and (9) converge to zero as $\delta \to 0$, moreover, it is expected that $\tilde{p}(\delta) \asymp \pi(\delta)$. The asymptotic behaviour of $\pi(\delta)$, as $\delta \to 0$, is obvious after substituting Q by its expression via δ:

$$\pi(\delta) = \frac{\sqrt{3}\delta^2}{4(\tilde{z} - y_{on})}. \tag{11}$$

To investigate the asymptotic behaviour of (9), the explicit form of C^{-1} is of interest, since C^{-1} is included into the expression of $\tilde{p}(\delta)$ via $M(\delta, Z)$ and $S^2(\delta)$. Recall that the matrix C is of the structure represented by the $m \times m$ matrix

$$U = \begin{pmatrix} 1 & a & \dots & a \\ a & 1 & \dots & a \\ \dots & \dots & \dots & \dots \\ a & a & \dots & 1 \end{pmatrix}, \tag{12}$$

where $a \in \mathbb{R}$.

Lemma 1. *The following equality holds*

$$V = U^{-1} = \frac{1}{u} \begin{pmatrix} t & a & \dots & a \\ a & t & \dots & a \\ \dots & \dots & \dots & \dots \\ a & a & \dots & t \end{pmatrix}, \tag{13}$$

where $t = -(m-2)a - 1$, *and* $u = (m-1)a^2 - (m-2)a - 1$.

Proof. The statement is proved simply by computing the elements of $W = U \cdot V$:

$$w_{ii} = \left((m-1)a^2 + t\right)/u = \left((m-1)a^2 - (m-2)a - 1\right)/u = 1,$$

$$w_{ij} = \left(a + at + (m-2)a^2\right)/u = \left(a + a(-(m-2)a - 1) + (m-2)a^2\right)/u = 0,\ i \neq j.$$

Thus, we have proved that W is a unit matrix. □

Corollary 1. *The inverse correlation matrix* C^{-1} *is a* $(d+1) \times (d+1)$ *matrix of the structure, presented by formula (13), where* $t = -(d-1)\exp(-c\delta^2) - 1$, *and* $u = d\exp(-2c\delta^2) - (d-1)\exp(-c\delta^2) - 1 = (d\exp(-c\delta^2) + 1)(\exp(-c\delta^2) - 1)$.

In order to investigate the convergence of $\tilde{p}(\delta)$ to 0, as $\delta \to 0$, we start from the convergence of its constituent parts, $M(\delta, Z)$ and $S^2(\delta)$.

The substitution of C^{-1} in (6) by its expression, defined in Corollary 1, yields

$$M(\delta, Z) = \mu + \exp\left(-c\delta^2 \frac{d}{2(d+1)}\right) \cdot I^T \cdot \frac{1}{u} \begin{pmatrix} t & a & \dots & a \\ a & t & \dots & a \\ \dots & \dots & \dots & \dots \\ a & a & \dots & t \end{pmatrix} \cdot (Z - \mu I) =$$

$$= \mu + \frac{1}{u} \exp\left(-c\delta^2 \frac{d}{2(d+1)}\right) \left(\exp(-c\delta^2) - 1\right) I^T \cdot (Z - \mu I) =$$

$$= \mu + \frac{\exp\left(-c\delta^2 \frac{d}{2(d+1)}\right) \left(\exp(-c\delta^2) - 1\right) \sum_{i=1}^{d+1} (z_i - \mu)}{(d\exp(-c\delta^2) + 1)(\exp(-c\delta^2) - 1)} =$$

$$= \mu + \frac{\exp\left(-c\delta^2 \frac{d}{2(d+1)}\right) \sum_{i=1}^{d+1} (z_i - \mu)}{d\exp(-c\delta^2) + 1}. \tag{14}$$

The expansion

$$\exp(-cx^2) = 1 - cx^2 + o(x^2), \tag{15}$$

applied to all the exponential terms in (14), gives the following asymptotic expression of $M(\delta, Z)$

$$M(\delta, Z) = \mu + \frac{(1 - c\delta^2 \frac{d}{2(d+1)} + o(\delta^2)) \sum_{i=1}^{d+1} (z_i - \mu)}{d + 1 - dc\delta^2 + o(\delta^2)} =$$
$$= \tilde{z} + o(\delta). \tag{16}$$

Similarly, the substitution of C^{-1} in (7) by its expression, defined in Corollary 1, yields

$$S^2(\delta) = \sigma^2 (1 - \exp\left(-c\delta^2 \frac{d}{d+1}\right) \cdot I^T \cdot \frac{1}{u} \begin{pmatrix} t & a & \dots & a \\ a & t & \dots & a \\ \dots & \dots & \dots & \dots \\ a & a & \dots & t \end{pmatrix} \cdot I =$$

$$= \sigma^2 (1 - \frac{1}{u} \exp\left(-c\delta^2 \frac{d}{d+1}\right) \left(-(d+1)((d-1)\exp(-c\delta^2) + 1) + (d^2 + d)\exp(-c\delta^2)\right) =$$

$$= \sigma^2 (1 - \frac{\exp\left(-c\delta^2 \frac{d}{d+1}\right)(d+1)\left(\exp(-c\delta^2) - 1\right)}{(d\exp(-c\delta^2) + 1)(\exp(-c\delta^2) - 1)} =$$

$$= \sigma^2 \left(1 - \frac{(d+1)\exp(-\frac{cd}{d+1}\delta^2)}{d\exp(-c\delta^2) + 1}\right). \tag{17}$$

The expansion

$$\exp(-cx^2) = 1 - cx^2 + \frac{1}{2}c^2x^4 + o(x^4), \tag{18}$$

applied to all the exponential terms in (17), yields the following asymptotic expression of $S^2(\delta)$

$$S^2(\delta) = \sigma^2 \frac{d\exp(-c\delta^2) - (d+1)\exp(-\frac{cd}{d+1}\delta^2) + 1}{d\exp(-c\delta^2) + 1} =$$

$$= \sigma^2 \frac{d(1 - c\delta^2 + \frac{1}{2}c^2\delta^4 + o(\delta^4)) - (d+1)(1 - \frac{cd}{d+1}\delta^2 + \frac{1}{2}\left(\frac{cd}{d+1}\right)^2 \delta^4 + o(\delta^4)) + 1}{d\exp(-c\delta^2) + 1}$$

$$= \sigma^2 \frac{d(-c\delta^2 + \frac{1}{2}c^2\delta^4 + o(\delta^4)) - (d+1)(-\frac{cd}{d+1}\delta^2 + \frac{1}{2}\left(\frac{cd}{d+1}\right)^2 \delta^4 + o(\delta^4))}{d\exp(-c\delta^2) + 1} =$$

$$= \sigma^2 \frac{-cd\delta^2 + \frac{1}{2}c^2 d\delta^4 + o(\delta^4) + \frac{cd(d+1)}{d+1}\delta^2 - \frac{1}{2}\left(\frac{cd}{d+1}\right)^2 (d+1)\delta^4 + o(\delta^4)}{d\exp(-c\delta^2)+1} =$$

$$= \sigma^2 \frac{(-cd+cd)\,\delta^2 + \frac{1}{2}\left(c^2 d - \frac{c^2 d^2}{d+1}\right)\delta^4 + o(\delta^4)}{d\exp(-c\delta^2)+1} =$$

$$= \sigma^2 \frac{\frac{c^2 d}{2(d+1)}\delta^4 + o(\delta^4)}{d\exp(-c\delta^2)+1} = \sigma^2 \frac{\frac{c^2 d}{2(d+1)}\delta^4 + o(\delta^4)}{d(1-c\delta^2+o(\delta^2))+1} =$$

$$= \sigma^2 \frac{\frac{c^2 d}{2(d+1)}\delta^4 + o(\delta^4)}{d+1+d(-c\delta^2+o(\delta^2))} = \sigma^2 \frac{\frac{c^2 d}{2(d+1)}\delta^4 + o(\delta^4)}{d+1+o(\delta)} =$$

$$= \sigma^2 c^2 \delta^4 \frac{d}{2(d+1)^2} + o(\delta^4). \tag{19}$$

The obtained assessments of asymptotic expressions (16) and (19) can be summarized as the following theorem.

Theorem 1. *The following equation is valid*

$$\tilde{p}(\delta) = \frac{\sigma c \delta^2}{(d+1)(\tilde{z}-y_{on})}\sqrt{\frac{d}{2}} + o(\delta^2). \tag{20}$$

Corollary 2. *In the case $d = 2$, the following relation of asymptotic equivalence is valid*

$$\tilde{p}(\delta) \sim \frac{4\sqrt{3}}{9}\sigma c \pi(\delta). \tag{21}$$

5 Conclusions

Two criteria used for justification of the construction of global optimization algorithms, based on simplicial statistical models, are considered. The first criterion refers to the improvement probability at the current optimization step, and the other is defined by a computationally simpler formula, but without a theoretical background, and is restricted to bi-variate problems. In the present paper, the asymptotic equivalence of both criteria is shown for the contracting simplices. The obtained result not only theoretically substantiates the application of the computationally simple criterion in the bi-variate case, but also supports its extension to higher dimensions.

Acknowledgements This research was funded by a grant (No. MIP-051/2014) from the Research Council of Lithuania.

References

1. Baritompa, W.: Customizing methods for global optimization: a geometric viewpoint. J. Glob. Optim. **3**(2), 193–212 (1993)
2. Calvin, J.M., Žilinskas, A.: On a global optimization algorithm for bivariate smooth functions. J. Optim. Theory Appl. **163**, 528–547 (2014)
3. Clausen, J., Žilinskas, A.: Global optimization by means of branch and bound with simplex based covering. Comput. Math. Appl. **44**, 943–995 (2002)
4. Fishburn, P.: Utility Theory for Decision Making. Wiley, New York (1970)
5. Horst, R., Pardalos, P., Thoai, N.: Introduction to Global Optimization. Kluwer Academic Publishers, New York (2007)
6. Huyer, W., Neumaier, A.: Global optimization by multi-level coordinate search. J. Glob. Optim. **14**, 331–355 (1999)
7. Kvasov, D., Sergeyev, Y.: Lipschitz gradients for global optimization in a one-point-based partitioning scheme. J. Comput. Appl. Math. **236**(16), 4042–4054 (2012)
8. Liuzzi, G., Lucidi, S., Picciali, V.: A partition-based global optimization algorithm. J. Glob. Optim. **48**(1), 113–128 (2010)
9. Mockus, J.: Bayesian Approach to Global Optimization. Kluwer Academic Publishers, Dordrecht (1988)
10. Pinter, J.: Global Optimization in Action. Kluwer Academic Publisher, Dordrecht (1996)
11. Sergeyev, Y., Kvasov, D.: Diagonal Global Optimization Methods. Fizmatlit, Moscow (2008, in Russian)
12. Sergeyev, Y., Strongin, R., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. Springer, New York (2013)
13. Strongin, R.G., Sergeyev, Y.D.: Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms. Kluwer Academic Publishers, Dordrecht (2000)
14. Törn, A., Žilinskas, A.: Global Optimization. Lecture Notes in Computer Science, vol. 350, Springer-Verlag, Berlin, pp. 1–255 (1989)
15. Wood, G.: Bisection global optimization methods. In: Floudas, C., Pardalos, P. (eds.) Encyclopedia of Optimization, pp. 186–189. Kluwer Academic Publisher, Dodrecht (2001)
16. Zhigljavsky, A., Žilinskas, A.: Stochastic Global Optimization. Springer, Berlin (2008)
17. Žilinskas, A.: Axiomatic approach to statistical models and their use in multimodal optimization theory. Math. Program. **22**, 104–116 (1982)
18. Žilinskas, A.: Axiomatic characterization of a global optimization algorithm and investigation of its search strategies. Oper. Res. Lett. **4**, 35–39 (1985)
19. Žilinskas, A.: Statistical models for global optimization by means of select and clone. Optimization **48**, 117–135 (2000)
20. Žilinskas, A.: On the statistical models-based multi-objective optimization. In: Butenko, S., Rassias, T., Floudas, C. (eds.) Optimization in Science and Engineering: In Honor of the 60th Birthday of Panos M. Pardalos, pp. 597–610. Springer Science+Business Media, New York (2014)
21. Žilinskas, A., Žilinskas, J.: Global optimization based on a statistical model and simplicial partitioning. Comput. Math. Appl. **44**, 957–967 (2002)

Advanced Statistical Tools for Modelling of Composition and Processing Parameters for Alloy Development

Greg Zrazhevsky, Alex Golodnikov, Stan Uryasev, and Alex Zrazhevsky

Abstract The paper presents new statistical approaches for modeling highly variable mechanical properties and screening specimens in development of new materials. Particularly, for steels, Charpy V-Notch (CVN) exhibits substantial scatter which complicates prediction of impact toughness. The paper proposes to use Conditional Value-at-Risk (CVaR) for screening specimens with respect to CVN. Two approaches to estimation of CVaR are discussed. The first approach is based on linear regression coming from the Mixed-Quantile Quadrangle, and the second approach builds CVN distribution with percentile regression, and then directly calculates CVaR. The accuracy of estimated CVaR is assessed with some variant of the coefficient of multiple determination. We estimated discrepancy between estimates derived by two approaches with the Mean Absolute Percentage error. We compared VaR and CVaR risk measures in the screening process. We proposed a modified procedure for ranking specimens, which takes into account the uncertainty in estimates of CVaR.

Keywords Steel • Toughness • Statistical Modeling • CVaR • Screening • Samples

G. Zrazhevsky (✉)
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
e-mail: zgrig@univ.kiev.ua

A. Golodnikov
Institute of Cybernetics of NAS of Ukraine, Kyiv, Ukraine

S. Uryasev
University of Florida, Gainesville, FL, USA

A. Zrazhevsky
American Optimal Decisions, Inc., Gainesville, FL, USA

A. Migdalas, A. Karakitsiou (eds.), *Optimization, Control, and Applications in the Information Age*, Springer Proceedings in Mathematics & Statistics 130,
DOI 10.1007/978-3-319-18567-5_21

1 Introduction

Development of new steels is an extremely costly and time-consuming process. The process involves two main stages: (1) deciding on chemical composition and processing parameters of the steel, based on previous experiences; (2) for the suggested composition of steel, production and testing of trial commercial specimens (see some description of the test of steel specimen at this link[1]). This testing process generates data on important mechanical characteristics of new steels such as yield tensile strength, elongation, and impact toughness (Charpy V-notch—CVN). The CVN impact test is designed to provide a measure of metal resistance to fast fracture in the presence of a flaw or notch. It has been used extensively in mechanical testing of steel products, in research, and in procurement specifications for over three decades. These mechanical characteristics are the basis for evaluation of obtained experimental specimens and selection of steels with the best properties for further development and more extensive testing. The selection of experimental specimens is not a trivial task. The experimental values of the mechanical characteristics in the selection process are not reliable, since they are random by its nature and may significantly depend on non-controlled conditions of physical experiments. The first stage in the development process can be done with statistical models such as ordinary linear regression model predicting underlying mean values of corresponding mechanical characteristics. While the tensile strength can be predicted with reasonable accuracy [1], the prediction of impact toughness is a much more difficult problem because experimental CVN data exhibit substantial scatter. The Charpy test does not provide a measure of an invariant material property, and CVN values depend on many parameters, including specimen geometry, stress distribution around the notch, and microstructural inhomogeneities around the notch tip. More on the CVN test, including the reasons behind the scatter and statistical aspects of this type of data analysis, can be found in [2–6]. Creating alloys with the best CVN values, therefore, results in multiple specimens for each experimental condition, leading to complex and expensive experimental programs.

To overcome this difficulty of predicting CVN, paper [1] suggested to use quantile regression, a nonparametric generalization of the ordinary least square regression introduced by Koenker and Bassett [7]. This technique predicts any given quantile (or percentile) of the distribution of CVN, rather than a single mean value (as in standard mean square regression). The quantile regression imposes minimal distributional assumptions on the data (the response is not required to be normal; data may be heteroscedastic, that is the variability in the response can vary for different values of the explanatory variables). Quantile regression combines results of measurement of the same dependent variable (CVN) that were collected from different specimens. Paper [1] used the quantile regression for predicting 20 % of CVN in the screening phase to assure that a specimen satisfies the toughness requirement (i.e., model-predicted 20 % value is higher than specified threshold).

[1] http://theconstructor.org/structural-engg/tensile-test-on-mild-steel-specimen/3514/.

In financial risk management, quantile, called the Value-at-Risk (VaR), is used to estimate tails of distributions. However, in recent years, Conditional Value-at-Risk (CVaR) is frequently used instead of VaR. Risk measure VaR provides a lower bound for the right tail of the distribution. Therefore, VaR does not measure the outcomes which of the most concern. In contrast to VaR, risk measure CVaR is an average of values upper VaR when right tail is estimated. More on the VaR, CVaR, and quantile regression can be found in [8–14].

This paper expands the approach described in [1] and considers CVaR instead of VaR for the estimating the right tail of CVN distribution. Simulation results presented in the paper are based on real-life data set described in [1]. This data set includes alloy chemical composition, plate thickness, and processing parameters (treating and ageing temperatures) for 34 specimens. Apart from these parameters, there correspond also to each specimen three values of CVN at $-84\,^{\circ}\mathrm{C}$.

Section 2 outlines the Risk Quadrangle [8] theory. Within this theory quantile (VaR) regression, which was pioneered in statistics by Koenker and Bassett [7], can be estimated using linear regression by minimizing Koenker–Bassett error (see [8]). CVaR is presented as mixed quantile (mixed VaR), and can be estimated using Linear Regression by minimizing Rockafellar error (see [8]). Thus, the first approach to estimation of CVaR is based on a Mixed-Quantile-Based Quadrangle.

Section 3 proposes the second approach to estimation of CVaR, which first generates sample of large size and builds CVN distribution, and then directly calculates CVaR of CVN. Accuracy of generated CVN distribution is assessed by using quantile regression equivalent of the coefficient of multiple determination. This section compares numerical performance of two approaches to CVaR estimations: (1) mixed percentile regression based on Rockafellar error; (2) distribution built with percentile regression.

Section 4 analyzes probabilistic models of CVN for different specimens, and compares three rules of specimens screening: (1) rule suggested in [3], which is based on the average and the 20th VaR of the CVN distribution; (2) rule suggested in [1] which is based only on the 20th VaR of the CVN distribution; (3) rule which is based only on the 20th CVaR of the CVN distribution. This section compares also performance of two approaches to estimation of CVaR in the process of screening specimens with respect to CVN.

Section 5 investigates precision of CVaR estimation based on distribution built with percentile regression. This section proposes a modified procedure for screening specimens, which takes into consideration uncertainty in estimates of CVaR.

2 Percentile and Mixed Percentile Regression

The Risk Quadrangle [8] theory defines groups of stochastic functionals called Quadrangles. Every quadrangle contains so-called Risk, Deviation, Error and Regret (negative utility). These elements of quadrangle are linked by so-called Statistics functional.

CVaR and VaR are elements of *Percentile Quadrangle*, in particular, CVaR is Risk and VaR is Statistics in this quadrangle (see [8]). Quadrangle is named after its Statistics, in this case Statistics is percentile (VaR). The Koenker–Bassett error is the Error in Percentile Quadrangle. Therefore, percentile (VaR) can be estimated using Linear Regression by minimizing Koenker–Bassett error (see [8]).

This section considers also *Mixed Percentile Quadrangle*. Mixed Percentile is Statistics and Rockafellar error is Error in *Mixed Percentile Quadrangle*. Mixed percentile (mixed VaR) can be estimated using Linear Regression by minimizing Rockafellar error (see [8]). CVaR for discrete distribution can be presented as Mixed VaR. Therefore, CVaR is Statistics in Mixed Percentile Quadrangle. It is interesting to observe that CVaR is Risk in Percentile Quadrangle and Statistics in Mixed Percentile Quadrangle.

Let us explain described concepts with exact mathematical terms. Let X be a random cost; $X_i(x),\ i = 0,1,\ldots,m$, is a family of random costs depending on a decision vector $x = (x_1,\ldots,x_n)$ belonging to a subset S. Measure of risk $\mathbf{R}$ aggregates the overall uncertain cost in X into a single numerical value $\mathbf{R}(X)$. This measure is used to model the statement "X adequately $\leq C$" by the inequality $\mathbf{R}(X) \leq C$.

Consider a family of random costs $X_i(x)$, $i = 0,1,\ldots,m$, depending on a decision vector $x = (x_1,\ldots,x_n)$ belonging to a subset S. A potential aim in choosing x from S would be to keep the random variable $X_i(x)$ adequately $\leq c_i$ for $i = 1,\ldots,m$, while achieving the lowest c_0 such that $X_0(x)$ is adequately $\leq c_0$. The way "adequately" could have different meaning for different i, and the notion of a risk measure addresses this issue. A selection of risk measure $\mathbf{R}_i$ that pins down the intended sense of "adequately" in each case leads to a optimization problem having the form

$$\text{choose } x \in S \text{ to minimize } \mathbf{R}_0(X_0(x)) \text{ subject to } \mathbf{R}_i(X_i(x)) \leq c_i \text{ for } i = 1,\ldots,m.$$

A measure of deviation $\mathbf{D}$ deals with uncertainty in a random variable X quantifying its nonconstancy. Thus $\mathbf{D}(X)$ is a generalization of the standard deviation $\sigma(X)$. Consideration of nonstandard measures of deviation in place of standard deviation is motivated by their ability to capture "heavy tail behavior" in probability distributions.

A measure of regret, $\mathbf{v}$, is introduced to quantify the net displeasure $\mathbf{v}(X)$ perceived in the potential mix of outcomes of random "costs" X. Regret comes up in penalty approaches to constraints in stochastic optimization and, in mirror image, corresponds to measure of "utility" $\mathbf{U}$ in a context of gains Y instead of losses X (which is typical in economics: $\mathbf{v}(X) = -\mathbf{U}(-X),\ \mathbf{U}(Y) = -\mathbf{v}(-Y)$). In applying U to Y the last is considered not as absolute gain but gain relative to some threshold, e.g., $Y = Y_0 - B$ where Y_0 is absolute gain and B is a benchmark.

A measure of error, ε, assigns to a random variable X a value $\varepsilon(X)$ that quantifies the nonzeroness in X. Classical examples are the norms

$$\|X\|_1 = E|X|, \quad \|X\|_p = [E|X|^p]^{1/p} \text{ for } p \in (1,\infty), \quad \|X\|_\infty = \sup|X|.$$

Given an error measure ε and a random variable X, one can look for a constant C nearest to X in the sense of minimizing $\varepsilon(X-C)$. The resulting minimum "ε-distance," denoted by $\boldsymbol{D}(X)$, is a deviation measure (see [8]). The C value in the minimum, denoted by $\mathbf{S}(X)$, can be called the "statistic" associated with X by ε. The case $\varepsilon(X) = \|X\|_2$ produces $\mathbf{S}(X) = EX$ and $\mathbf{D}(X) = \sigma(X)$. The generation of a particular deviation measure $\mathbf{D}$ and statistic $\mathbf{S}$ from an error measure ε has implications for statistical estimation in the sense of generalized regression.

Regression is a way of approximating a random variable Y by a function $f(X_1,\ldots,X_n)$ of one or more random variables X_j. The regression evaluates with error measure ε how far the random difference $Z_f = Y - f(X_1,\ldots,X_n)$ is from 0. For an error ε and a collection $\mathbf{C}$ of regression functions f, the basic problem of regression for Y with respect to $X_1,\ldots,X_n$ is to

$$\text{minimize } \varepsilon(Z_f) \text{ over } f \in \mathbf{C}, \text{ where } Z_f = Y - f(X_1,\ldots,X_n). \tag{1}$$

To illustrate richness of the quadrangle scheme and the interrelationships between quadrangle objects, consider the *Quantile Quadrangle*, and a *Mixed Quantile Quadrangle* [8].

The *Quantile Quadrangle* combines quantile statistics with concepts from risk. By tying "Conditional Value-at-Risk", on the optimization side, to Quantile Regression as pioneered in statistics by Koenker and Bassett [7], it underscores a relationship that might go unrecognized without the risk quadrangle scheme.

Let us consider the (cumulative) distribution function $F_X(x) = P\{X \le x\}$ of a random variable X and the quantile values associated with it. If, for a probability level $\alpha \in (0,1)$, there is a unique x such that $F_X(x) = \alpha$, then x, by definition, is the α-quantile $q_\alpha(X)$. In general, however, there are two values to consider as extremes:

$$q_\alpha^+(X) = \inf\{x | F_X(x) > \alpha\}, \qquad q_\alpha^-(X) = \sup\{x | F_X(x) < \alpha\}.$$

It is customary, when these differ, to take the lower value as the α-quantile, noting that, because F_X is right-continuous, this is the lowest x such that $F_X(x) = \alpha$. Consider the entire interval between the two competing values as the quantile,

$$q_\alpha(X) = \left[q_\alpha^-(X), q_\alpha^+(X)\right].$$

In finance, the Value-at-Risk term is used for quantile, and upper VaR $\mathrm{VaR}_\alpha^+(X) = q_\alpha^+(X)$ along with a lower VaR $\mathrm{VaR}_\alpha^-(X) = q_\alpha^-(X)$, and the VaR interval $\mathrm{VaR}_\alpha(X) = \left[\mathrm{VaR}_\alpha^-(X), \mathrm{VaR}_\alpha^+(X)\right]$ is identical to the quantile interval $q_\alpha(X)$.

Besides VaR, the example coming under consideration involves the CVaR of X at level $\alpha \in (0,1)$, defined by

$$\mathrm{CVaR}_\alpha(X) = \text{ expectation of } X \text{ in its } \alpha\text{-tail},$$

which is also expressed by

$$\mathrm{CVaR}_\alpha(X) = \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_\tau(X)\,d\tau. \tag{2}$$

Conditional Value-at-Risk $\mathrm{CVaR}_\alpha(X)$ is also called in [8] by superquantile $\overline{q}_\alpha(X)$.

Let $X_+ = \max\{0, X\}$, $X_- = \max\{0, -X\}$, $X = X_+ - X_-$.

A *Quantile Quadrangle* has the following elements:

- statistic $\mathbf{S}(X) = \mathrm{VaR}_\alpha(X) = q_\alpha(X)$ = quantile;
- risk $\mathbf{R}(X) = \mathrm{CVaR}_\alpha(X) = \overline{q}_\alpha(X)$= superquantile;
- deviation $\mathbf{D}(X) = \mathrm{CVaR}_\alpha(X - EX) = \overline{q}_\alpha(X - EX)$ = superquantile-deviation;
- regret $\mathbf{v}(X) = \dfrac{1}{1-\alpha} EX_+$ = average absolute loss, scaled;
- error $\varepsilon(X) = E\left[\dfrac{\alpha}{1-\alpha} X_+ + X_-\right]$ = normalized Koenker–Basset error.

The original Koenker–Bassett Error expression differs from the normalized Koenker-Basset error in *Quantile Quadrangle* by a positive factor. In order to build regression function $f(X_1, \ldots, X_n)$ which approximates percentile of random variable Y one should solve the optimization problem (1) with normalized Koenker–Basset error.

Consider the case when random variable Y is approximated by the linear function of a vector of $K+1$ explanatory variables $\mathbf{x}' = [1, x_1, \ldots, x_K]$, $Y = \mathbf{x}'\beta + \delta$, where $\beta' = [\beta_0, \beta_1, \ldots, \beta_K]$. The δ is zero-mean random term that accounts for the surplus variability or scatter in Y that cannot be explained by explanatory variables $x_1, \ldots, x_K$. The cumulative effects of unmeasured and/or unforeseen variables are usually lumped into the stochastic δ term.

Let Y denote the logarithm of CVN, ln(CVN), and $Y_1, \ldots, Y_n$ are observations of the random variable Y at points $\mathbf{x}_i' = \left[1, x_1^i, \ldots, x_K^i\right]$, $i = 1, \ldots, n$. Then estimates of coefficients of the α-th quantile regression function can be found by minimizing the normalized Koenker–Basset error

$$\frac{1}{n}\left\{\sum_{i:Y_i \geq \mathbf{x}_i'\beta_\alpha} \frac{\alpha}{(1-\alpha)} \left|Y_i - \mathbf{x}_i'\beta_\alpha\right| + \sum_{i:Y_i \leq \mathbf{x}_i'\beta_\alpha} \left|Y_i - \mathbf{x}_i'\beta_\alpha\right|\right\}. \tag{3}$$

A *Mixed-Quantile Quadrangle* has the following elements for confidence levels $\alpha_k \in (0,1)$ and weights $\lambda_k > 0$, $\sum_{k=1}^{r} \lambda_k = 1$:

- statistic $\mathbf{S}(X) = \sum_{k=1}^{r} \lambda_k q_{\alpha_k}(X) = \sum_{k=1}^{r} \lambda_k \mathrm{VaR}_{\alpha_k}(X)$ = mixed quantile;
- risk $\mathbf{R}(X) = \sum_{k=1}^{r} \lambda_k \overline{q}_{\alpha_k}(X) = \sum_{k=1}^{r} \lambda_k \mathrm{CVaR}_{\alpha_k}(X)$ = mixed superquantile;
- deviation $\mathbf{D}(X) = \sum_{k=1}^{r} \lambda_k \overline{q}_{\alpha_k}(X - EX) = \sum_{k=1}^{r} \lambda_k \mathrm{CVaR}_{\alpha_k}(X - EX)$ = corresponding mixture of superquantile deviations;

- regret $\mathbf{v}(X) = \min\limits_{B_1,\ldots,B_r} \left\{ \sum\limits_{k=1}^{r} \lambda_k \mathbf{v}_{\alpha_k}(X - B_k) \,\middle|\, \sum\limits_{k=1}^{r} \lambda_k B_k = 0 \right\}$ = derived balance of the regrets $\mathbf{v}_{\alpha_k}(X) = \dfrac{1}{1-\alpha_k} EX_+$;
- error $\varepsilon(X) = \min\limits_{B_1,\ldots,B_r} \left\{ \sum\limits_{k=1}^{r} \lambda_k \varepsilon_{\alpha_k}(X - B_k) \,\middle|\, \sum\limits_{k=1}^{r} \lambda_k B_k = 0 \right\}$ = Rockafellar error function,

where $\varepsilon_{\alpha_k}(X) = E\left[\dfrac{\alpha_k}{1-\alpha_k} X_+ + X_-\right]$ = normalized Koenker–Basset error with α_k.

Relationship between $\mathrm{CVaR}_\alpha(Y)$ and *Mixed-Quantile Quadrangle* is established by the formula (2). Classical numerical integration uses a finite subdivision of the interval $[\alpha, 1]$ and replaces the integrand in (2) by a nearby step function or piecewise linear function based on the quantiles marking that subdivision. It is easy to see that the value of the integral for that approximated integrand is actually a mixed quantile expression. Thus for confidence levels $\alpha_r \in (\alpha, 1)$, $r = 1, \ldots, R$, and weights $\lambda_r > 0$, $\sum\limits_{r=1}^{R} \lambda_r = 1$, $\mathrm{CVaR}_\alpha(Y)$ in (2) can be approximated by the mixed quantile

$$\mathrm{CVaR}_\alpha(Y) \approx \sum_{r=1}^{R} \lambda_r \mathrm{VaR}_{\alpha_r}(Y). \tag{4}$$

CVaR regression function is a generalization of the mixed quantile (4) to the case when Y is a linear function of a vector of $K+1$ explanatory variables $\mathbf{x}' = [1, x_1, \ldots, x_K]$ plus random error, $Y = \mathbf{x}'\beta + \varepsilon$, where $\beta' = [\beta_0, \beta_1, \ldots, \beta_K]$.

It is estimated by minimizing Rockafellar error function with

$$\varepsilon_{\alpha_k}(X) = E\left[\frac{\alpha_k}{1-\alpha_k} X_+ + X_-\right] =$$

$$= \frac{1}{n} \left\{ \sum_{i:Y_i \geq \mathbf{x}_i'\beta_\alpha} \frac{\alpha_k}{(1-\alpha_k)} \left| Y_i - \mathbf{x}_i'\beta_\alpha \right| + \sum_{i:Y_i \leq \mathbf{x}_i'\beta_\alpha} \left| Y_i - \mathbf{x}_i'\beta_\alpha \right| \right\}.$$

An important issue in regression with many independent variables is how to choose a subset of variables so that a large portion of variability of Y is explained by these few x variables. In the ordinary least squares model, one can measure the proportion of variability through a quantity known as R^2; e.g., $R^2 = 90\,\%$ means that only $10\,\%$ of the variation in Y cannot be accounted for by the x variables. In the quantile regression, criteria R^2 is not applicable. To assess the goodness of fit, Koenker and Machado [11] introduced a quantile regression equivalent of the coefficient of multiple determination, $R^1(\alpha)$, which measures the proportion of the variability in the response that is accounted for by the fitted α-th quantile surface.

In order to exclude variables that contribute little to the explanation of Y, we applied the stepwise variable selection method. As in its standard least squares regression counterpart, this procedure adds and removes explanatory variables alternately from the model using significance tests, eventually converging to a final subset of variables.

3 Building CVN Distributions and Estimation of CVaR for Specimens

In Sect. 2 we considered methods for estimating VaR and CVaR using the *Quantile Quadrangle*, and a *Mixed Quantile Quadrangle*. This section suggests an alternative approach to estimation of VaR and CVaR for CVN distributions. The idea of the approach is to build CVN distribution for each specimen. Then with these distributions we estimate VaR and CVaR.

With quantile regression we can estimate any quantile of the CVN distribution. Moreover, quantile regression can estimate the whole distribution by estimating all quantiles (or at least estimating quantiles with some fine grid). The idea here is to use information over a large number of quantiles to reduce the approximation error and enhance the accuracy of the estimated whole distribution for each specimen. By definition, quantile is the inverse function of the corresponding cumulative probability distribution function.

We constructed distribution of CVN for each specimen using the following procedure, repeated 10,000 times:

1. Draw a random value from the Uniform $(0,1)$ distribution and treat it as a probability level α;
2. Build quantile regression model for this probability level;
3. For each specimen calculate quantile by substituting specimen-specific composition and processing parameters into quantile regression model with parameter α and treat it as a random realization of specimen-specific CVN random value.

Thus, for each specimen we generated specimen-specific large sample of CVN values, which is used for building empirical specimen-specific CVN distribution.

This procedure together with quantile regression techniques is the tool, which transforms available information about all produced and tested trial commercial specimens (chemical composition, processing parameters, and CVN) in the distribution of CVN for each specimen. The more values of probability levels α are involved in this procedure, the more accurate is the transformation. Thus our current-state of knowledge about interrelation between chemical composition, processing parameters, and CVN values is completely presented in form of the CVN distribution. In this sense for large sample size (10,000 or more) we can consider such distributions as a good probabilistic models of CVN for corresponding specimen.

Table 1 Accuracy of different regions of the generated CVN distribution

Range of α	Range of coefficients $R^1(\alpha)$
0.050–0.250	0.4219–0.4424
0.251–0.350	0.3488–0.4400
0.351–0.533	0.4400–0.4871
0.534–0.665	0.3400–0.3796
0.666–0.750	0.3015–0.3399
0.751–0.832	0.2500–0.3015
0.833–0.928	0.2001–0.2500
0.929–0.980	0.1954–0.3765

Accuracy of generated CVN distribution may be assessed by using quantile regression equivalent of the coefficient of multiple determination, $R^1(\alpha)$, which measures the proportion of the "variability" in the response that is accounted for by the fitted α-th quantile surface.

Table 1 shows that for the wide range of probability levels ($0.05 \leq \alpha \leq 0.832$) the quantile regression functions capture more than 25 % of the variability in the response that is accounted for by the fitted α-th quantile surface. The most accurate portion of the generated CVN distribution is in the range of probability levels $0.351 \leq \alpha \leq 0.533$. Quantile regression functions corresponding to this range of α account for 44–48.71 % of the response variability.

For constructed CVN distributions we can determine for each specimen the following characteristics: average, α-th quantile, and α-th CVaR. We used these characteristics in the process of screening specimens with respect to CVN.

Let us compare numerical performance of two approaches to CVaR estimations: (1) mixed percentile regression based on Rockafellar error; (2) distribution built with percentile regression. We transformed specimen-specific CVN distributions into corresponding distributions of ln(CVN) and calculated α-th CVaR for each distribution. Let J denote the total number of specimens, and let $\mathrm{CVaR}_\alpha(\ln(\mathrm{CVN}_j))$ be α-th CVaR of ln(CVN) for j-th specimen, $j = 1,\ldots,J$. Suppose that $\mathrm{CVaR}_\alpha(\ln(\mathrm{CVN})) = \mathbf{x}'\beta$ is CVaR regression function found by minimizing Rockafellar error function, $\mathbf{x}' = [1, x_1, \ldots, x_K]$ are explanatory variables, $\beta' = [\beta_0, \beta_1, \ldots, \beta_K]$ are coefficients of CVaR regression function. Here $\mathbf{x}'_j = \left[1, x_1^j, \ldots, x_K^j\right]$ are values of explanatory variables for j-th specimen, $j = 1,\ldots,J$. Then $\mathbf{x}'_j\beta$ is estimate of α-th CVaR of ln(CVN) derived from CVaR regression.

Discrepancy between estimates based on Rockafellar error and on distribution built with percentile regression can be measured by Mean Absolute Error (MAE),

$$\mathrm{MAE} = \frac{1}{J}\sum_{j=1}^{J}\left|\mathrm{CVaR}_\alpha(\ln(\mathrm{CVN}_j)) - \mathbf{x}'_j\beta\right|, \tag{5}$$

Table 2 Average discrepancy between estimates based on Rockafellar error and on distribution built with percentile regression for equal weights $\lambda_1, \ldots, \lambda_R$

Number of VaRs in Mixed Quantile, R	MAE	MAPE (%)
5	0.1519	4.04
10	0.1643	4.31
15	0.1631	4.27
20	0.167	4.36
25	0.1680	4.38
30	0.1754	4.55

or Mean Absolute Percentage Error (MAPE),

$$\text{MAPE} = \frac{1}{J}\sum_{j=1}^{J}\left|\frac{\text{CVaR}_\alpha(\ln(\text{CVN}_j)) - \mathbf{x}_j'\beta}{\text{CVaR}_\alpha(\ln(\text{CVN}_j))}\right|. \tag{6}$$

Using original data with three observed values of CVN for each specimen, we built 20 %-CVaR regression functions for different values of VaRs in the mixed quantile (4), with equal distance between adjacent values of confidence levels $\alpha_r \in (0.2, 1)$, $r = 1, \ldots, R$, and with equal values of weights $\lambda_1, \ldots, \lambda_R$. Average discrepancies between these estimates are presented in Table 2.

Analyzing Table 2 we draw the following conclusions:

1. Two approaches to CVaR estimations (based on Rockafellar error and on distribution built with percentile regression) provide on average similar estimates of $\text{CVaR}_{0.2}(\ln(\text{CVN}))$ for all specimens. The average discrepancy between these estimates does not exceed 4.55 % in the conducted numerical experiments.
2. This result only slightly depends on the number of VaRs in the mixed quantile. The average discrepancy between these estimates increases from 4.04 to 4.55 % with increase of number of VaRs in mixed quantile from 5 to 30.

We investigated also the case when for a given set of confidence levels $\alpha_r \in (0.2, 1)$, $r = 1, \ldots, R$, corresponding weights $\lambda_1, \ldots, \lambda_R$ are chosen to get the best approximation of $\text{CVaR}_\alpha(\ln(\text{CVN}_j))$ by the sum $\sum_{r=1}^{R}\lambda_r \text{VaR}_{\alpha_r}(\ln(\text{CVN}_j))$ in (4). For this purpose we minimized MAE of such approximation for all specimens

$$\min_{\lambda_1,\ldots,\lambda_R} \frac{1}{J}\sum_{j=1}^{J}\left|\text{CVaR}_\alpha(\ln(\text{CVN}_j)) - \sum_{r=1}^{R}\lambda_r \text{VaR}_{\alpha_r}(\ln(\text{CVN}_j))\right| \tag{7}$$

subject to

$$\sum_{r=1}^{R}\lambda_r = 1, \tag{8}$$

$$\lambda_r \geq 0, \quad r = 1, \ldots, R. \tag{9}$$

Table 3 Average discrepancy between estimates based on Rockafellar error and on distribution built with percentile regression with weights $\lambda_r^*, r = 1, \ldots, R$

Number of VaRs in Mixed Quantile, R	MAE	MAPE (%)
5	0.1422	3.74
10	0.1906	4.93
15	0.1763	4.54
20	0.1802	4.63
25	0.1798	4.63
30	0.1816	4.67

We solved optimization problem (7)–(9) for different number of VaRs in mixed quantile, $R = 5, 10, 15, 20, 25, 30$. For all values of r optimal solutions, λ_r^*, $r = 1, \ldots, R$, are not equal. For example, for $R = 5$, $\lambda_1^* = 0.390$, $\lambda_2^* = 0$, $\lambda_3^* = 0.198$, $\lambda_4^* = 0$, $\lambda_5^* = 0.412$.

Average discrepancies between these estimates for optimal values λ_r^*, $r = 1, \ldots, R$, are presented in the Table 3.

Comparing Tables 2 and 3 we conclude that goodness of fit of CVaR regression in cases when all weights are equal and when optimal values λ_r^*, $r = 1, \ldots, R$ are used is approximately the same.

4 Comparison of VaR, and CVaR Risk Measures in Screening Process

As described in [3], the alloy development process includes three rounds of laboratory melting, processing and testing of candidate steels. Round 1 involves screening of several exploratory compositions. Round 2 consists of additional laboratory testing of new heats for the best compositions from Round 1. The most promising steel from Round 2 (the prime candidate steel) is the subject of more extensive testing in Round 3, in which several heats are tested. The ranking and selection of best alloys are based on combinations of yield strength and CVN at $-84\,^{\circ}\mathrm{C}$.

In [1] we have shown that the tensile yield strength can be accurately predicted with ordinary linear regression. At the same time, we encountered difficulty in attempting to build a linear regression model for CVN at $-84\,^{\circ}\mathrm{C}$. Whereas each specimen is characterized by a single value of yield strength, three values are observed for CVN. This setting and the fact that there is a substantial variability in these values complicates the process of screening specimens with respect to CVN.

As specified in [3], an acceptable specimen must satisfy the following toughness requirement: the average of three values of CVN must be greater than specified threshold c, with no single value below c by more than some specified value d.

It is clear that this ranking criterion is not perfect, since it is based on a very small number of random observations (only three values of CVN). In fact, screening

with this rule is random and unstable. To overcome this shortcoming we (see [1]) screened specimens with respect to CVN with quantile (percentile) regression. Quantile regression involves CVN values from testing of several specimens (not only one specimen). Therefore, ranking criterion based on quantile regression is more stable.

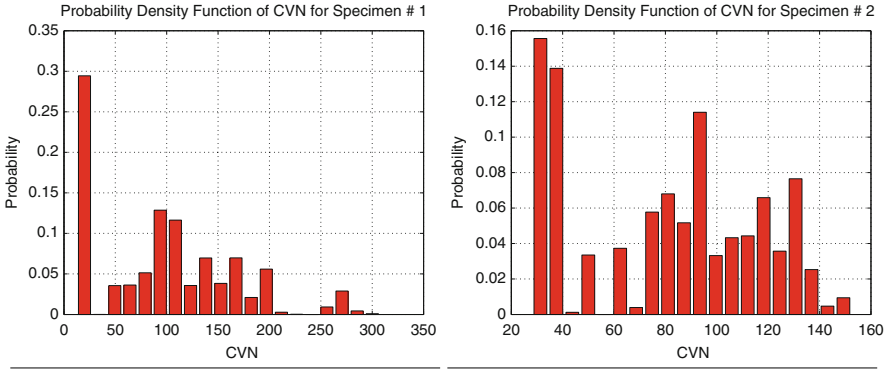
In order to determine which quantile corresponds to the smallest of the three values of CVN, we successively drew three random values from a standard normal distribution, each time selecting only the smallest value. This was repeated 10,000 times, producing an average value of 0.19. The interpretation in [1] was that the smallest of three such CVN values approximately corresponds to the 20th percentile of the CVN distribution. The ranking criterion in [1] was based solely on the 20th percentile of the CVN distribution.

Nevertheless, quantile regression does not evaluate the tail of CVN distribution. In contrast to quantile, the CVaR is an average of outcomes exceeding this quantile. Conceptually, CVaR is preferable to quantile for screening specimens with respect to CVN.

We analyzed probabilistic models of CVN for different specimens using the following rules of specimens screening:

1. Rule suggested in [3] which is based on the average and the 20th VaR of the CVN distribution;
2. Rule suggested in [1] which is based on the 20th VaR of the CVN distribution;
3. Rule which is based on the 20th CVaR of the CVN distribution.

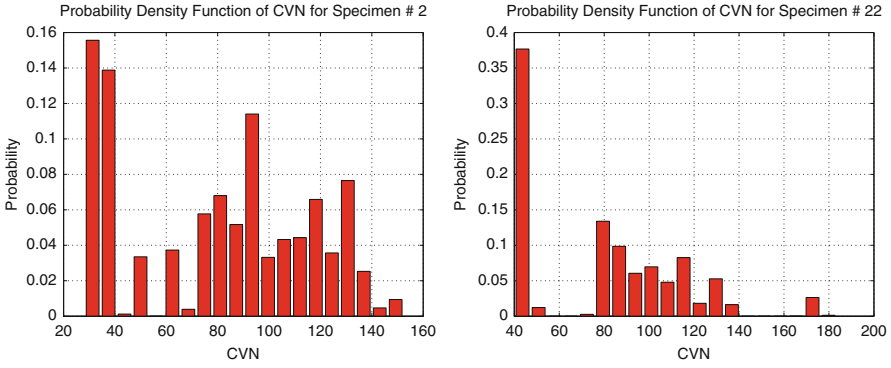
Figure 1 shows that specimen 2 has larger value of 20 % percentile (20 %–VaR) of the CVN distribution than the specimen 1, but smaller values of average and 20 % CVaR. Therefore, the screening rule 1 could not select the better specimen from these two specimens; rule 2 should classify the specimen 2 as better one than



specimen #	20%-VaR	20%-CVaR	Mean
1	19.21	117.55	97.1

specimen #	20%-VaR	20%-CVaR	Mean
2	35.37	91.75	79.63

Fig. 1 Probabilistic models for specimen 1 and 2, and their characteristics



specimen #	20%-VaR	20%-CVaR	Mean
2	35.37	91.75	79.63

specimen #	20%-VaR	20%-CVaR	Mean
22	45.00	89.42	80.13

Fig. 2 Probabilistic models for specimen 2, and 22, and their characteristics

specimen 1; rule 3 should classify the specimen 1 as better one than specimen 2. Since CVaR is preferable to quantile for screening specimens with respect to CVN, the correct classification provides only rule 3.

Figure 2 shows that the specimen 22 has larger value of 20 % percentile (20 %–VaR) and the mean of the CVN distribution than the specimen 2, but smaller value of 20 % CVaR. Therefore, the screening rules 1 and 2 should classify the specimen 22 as a better one than specimen 2, but rule 3 should classify the specimen 2 as a better one than specimen 22. Since CVaR is preferable to quantile for screening specimens with respect to CVN, the correct classification provides only rule 3.

These examples demonstrate that results of screening with rules 1, 2, and 3 contradict each other. Rule 3 which is based on the CVaR of the CVN distribution is more appropriate and it is recommended for screening process. We also emphasize that in regression and in probabilistic modeling, all available information should be used to increase accuracy of screening process. For instance, screening of specimens in Round 1 can use CVN distributions derived from data generated during this Round. According to observations from Round 1, these CVN distributions are considered as "correct". Round 2 generates additional information, which should be combined with information from Round 1 and utilized for derivation of new CVN distributions. These new CVN distributions are used for screening of specimens in Round 2. Finally, screening of specimens in Round 3 should use CVN distributions derived from data generated during three rounds.

We compared performance of the two approaches to estimation of CVaR in the process of screening with respect to CVN by using 33 pairs of specimens.

The first approach, which is based on a *Mixed Quantile Quadrangle*, utilizes original experimental data for direct estimating of 20 %-CVaR for $\ln(\text{CVN})$. Therefore, in screening two specimens with numbers j_1, and j_2 the first approach should compare two values $\text{CVaR}_{0.2}(\ln(\text{CVN}_{j_1}))$, and $\text{CVaR}_{0.2}(\ln(\text{CVN}_{j_2}))$, j_1, $j_2 = 1,\ldots,J$.

The second approach estimates 20 %-CVaR in two steps. At the first step, the original experimental data are used for building distributions of CVN for each specimen. At the second step we directly calculated 20 %-CVaR for each specimen using these CVN distributions. In screening, two specimens with numbers j_1, and j_2 the second approach should compare two values $\mathrm{CVaR}_{0.2}(\mathrm{CVN}_{j_1})$, and $\mathrm{CVaR}_{0.2}(\mathrm{CVN}_{j_2})$, $j_1, j_2 = 1, \ldots, J$.

Since logarithm is an increasing function, the inequalities $\mathrm{CVaR}_{0.2}(\mathrm{CVN}_{j_1}) > \mathrm{CVaR}_{0.2}(\mathrm{CVN}_{j_2})$, and $\mathrm{CVaR}_{0.2}(\ln(\mathrm{CVN}_{j_1})) > \mathrm{CVaR}_{0.2}(\ln(\mathrm{CVN}_{j_2}))$, are equivalent. Both approaches provide the same result only in screening of 20 pairs. In the 13 pairs results of screening were different. Despite the fact that the average discrepancy between estimates of $\mathrm{CVaR}_{0.2}(\ln(\mathrm{CVN}))$ obtained by using these two approaches do not exceed 4.55 % (see results in the Table 2), discrepancy in results of screening exceeds 39 %. Since the second approach uses 10,000 CVN values for each specimen, while the first approach uses only 3 values, we consider the second approach is more accurate.

5 Precision of Estimates of CVaR Derived From Small Samples

Section 3 describes procedure for building a CVN distribution for each specimen and determining characteristics of these distributions. The procedure is based on generating large samples of CVN values for each specimen by using quantile regressions. If size of the sample is large, it can be considered as a population corresponding to a specimen, and characteristics of CVN distribution such as average, 20 %-quantile, and 20 %-CVaR can be considered as "true" characteristics corresponding to the specimen. Therefore, these characteristics can be used in the screening process. We considered that samples containing 10,000 values can be classified as large, although for high precision, sample size 100,000 or 1,000,000 may be needed. For such large sample sizes, procedure of building CVN distributions is time consuming.

However, usually, samples of small and moderate size are used. In this case estimates of average, VaR, and CVaR derived from small samples (with size 100, 200, 500) may have large uncertainty, which should be taken into consideration in the screening process. This uncertainty can be quantitatively characterized by 90 %-th or 95 %-th confidence intervals. The "confidence interval probability" is the probability that the calculated confidence interval encompasses the true value of the population parameter (average, VaR, or CVaR). The size of the confidence interval, and the confidence levels provide information on the accuracy of the estimation.

In order to build confidence interval, we randomly sample from a population of size N to produce k new samples of fixed size n. Each sample consists of random realizations of random variables $\xi_1, \ldots, \xi_n$. Functions of these random variables, average, 20 % VaR, and 20 % CVaR, called statistics, are also random.

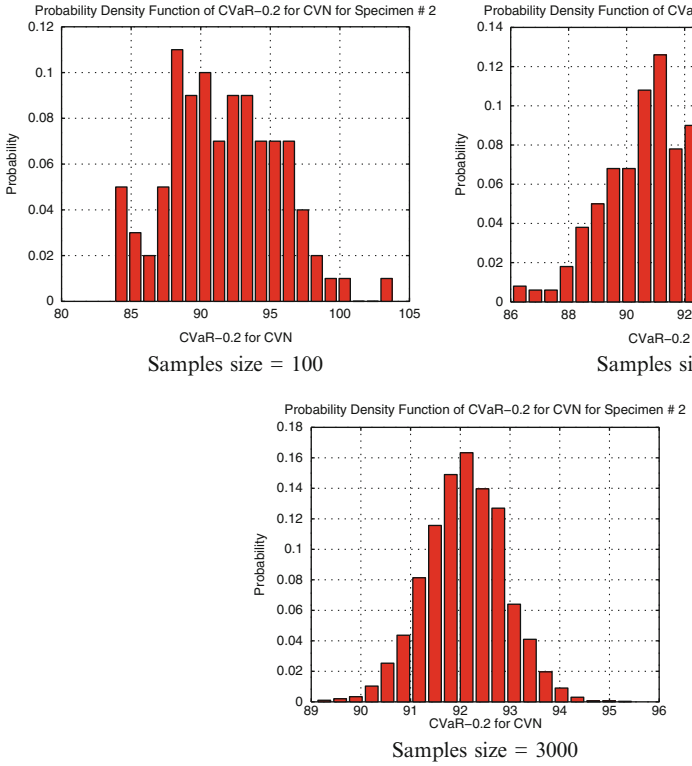


Fig. 3 Empirical sampling distributions of 20 % CVaR for CVN for Specimen 2 (number of generated samples = 10,000, sample sizes =100; 500; 3,000)

From each generated sample we calculated random realizations of these statistics. With a large number of new samples, $k = 10{,}000$, we generated empirical sampling distributions separately for average, 20 % VaR, and 20 % CVaR. For each of these empirical sampling distributions we determined average, 5 %- and 95 %-quantiles, which specify point estimate as well as lower and upper confidence limits for corresponding statistics.

In our calculations we used the following sampling parameters:

1. population size $N = 10{,}000$;
2. number of generated small samples $k = 10{,}000$;
3. sizes of small samples $n = 100, 200, 300, 500, 1{,}000, 3{,}000$;

Figures 3 and 4 show empirical sampling distributions of the population parameters (20 % CVaR, and 20 % VaR for CVN) in case when random samples of fixed sizes ($n = 100$, 500, and 3,000) were drawn from the population. Figure 3 demonstrates that sample 20 % CVaR distribution gradually converges to the normal distribution when sample size increases from 100 to 3,000. But this is not the case for 20 % VaR distributions (see Fig. 4).

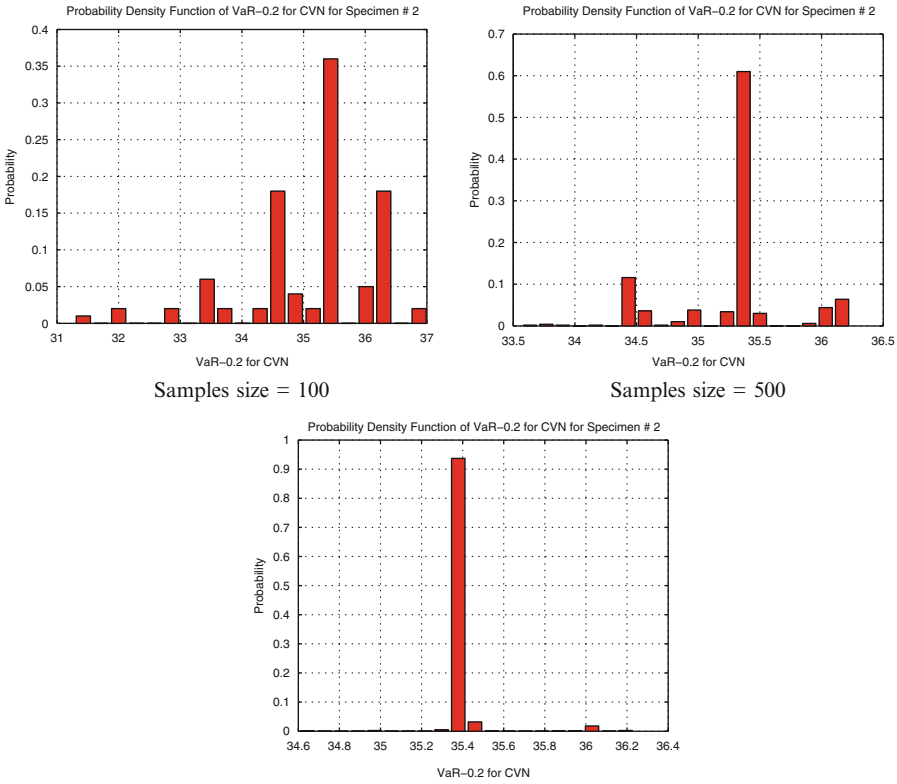


Fig. 4 Empirical sampling distributions of 20 % VaR for CVN for Specimen 2 (number of generated samples = 10,000, sample sizes =100; 500; 3,000)

Empirical sampling distributions of the population parameters were used for determining their point estimates, 90 % confidence intervals, and the accuracy of estimation calculated as the ratio of length of the confidence interval to the corresponding point estimate. The point estimate for a population parameter was determined as the mean of its empirical sampling distribution. Estimates of 20 % VaR for CVN distribution for Specimen 2 obtained for samples of size 100, 200, 300, 500, 1,000, and 3,000 are in Table 4.

The Table 4 shows the rapid growth of accuracy of the 20 % VaR for CVN (Specimen 2) estimation while a sample size increases from 100 to 3,000. The best accuracy, 0.20 %, is achieved for sample size = 3,000, it is significantly better than for sample size = 1,000.

Similar results of estimation for 20 % CVaR for CVN (Specimen 2) are presented in Table 5.

Table 4 Estimates of 20 % VaR for CVN (Specimen 2)

Sample size	Point estimate	90 % Confidence interval		Accuracy of estimation (%)
		5 %-Quantiles	95 %-Quantiles	
100	35.30	33.43	36.65	9.11
200	35.28	33.81	36.23	6.86
300	35.31	34.44	36.23	5.07
500	35.34	34.44	36.23	5.06
1,000	35.36	34.61	36.05	4.10
3,000	35.39	35.37	35.44	0.20

Table 5 Estimates of 20 % CVaR for CVN (Specimen 2)

Sample size	Point estimate	90 % Confidence interval		Accuracy of estimation (%)
		5 %-Quantiles	95 %-Quantiles	
100	91.82	84.51	99.02	15.80
200	91.79	86.63	96.96	11.25
300	91.74	87.51	95.95	9.20
500	91.77	88.57	94.99	7.00
1,000	91.75	89.43	94.06	5.05
3,000	92.09	90.76	93.40	2.86

The Table 5 shows the moderate growth of accuracy of the 20 % CVaR for CVN (Specimen 2) estimation while a sample size increases from 100 to 3,000. However, the comparison of Tables 4 and 5 shows that estimates of 20 % VaR are more accurate than estimates of 20 % CVaR for all sample sizes. For instance, for samples size = 3,000 accuracy of estimation of 20 % CVaR is 10 times worse than accuracy of estimation of 20 % VaR. Therefore, in the screening procedure based on estimation of CVaR derived from samples of moderate sizes we have to use confidence intervals.

All values in the 90 % confidence interval for CVaR are plausible values for the CVaR with probability 0.9, whereas values outside the interval are rejected as plausible values for CVaR. Therefore, in screening process based on estimates of CVaR, which were derived from samples of moderate sizes, we should compare confidence intervals of two specimens instead of their point estimates as described in Sect. 4. If these confidence intervals do not overlap, these specimens are necessarily significantly different. In this case, the specimen, corresponding to the confidence interval with lower values, should be disqualified. However, if these specimens have overlapping confidence intervals, the screening procedure cannot select the best specimen. In this case, we should increase sample size and repeat the procedure of estimation of confidence interval.

For instance, consider confidence intervals for 20 %-CVaR, corresponding to specimen 2 and specimen 10, shown in Fig. 5.

Figure 5 shows that, if sample size equals 100, then these specimens have overlapping confidence intervals. In this case, the screening procedure cannot select

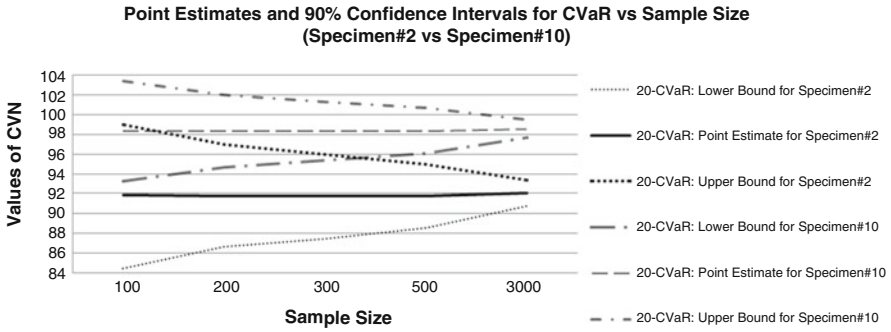


Fig. 5 Confidence intervals for 20 %-CVaR, corresponding to specimens 2 and 10, which were derived from 10,000 samples of fixed size 100, 200, 300, 500, and 1,000

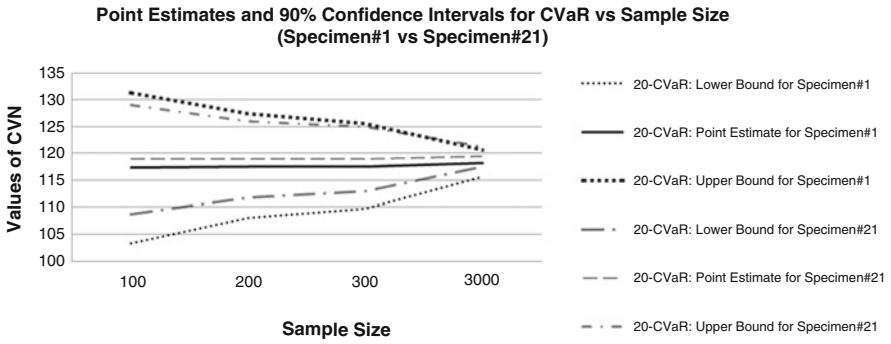


Fig. 6 Confidence intervals for 20 %-CVaR, corresponding to specimens 1 and 21, which were derived from 10,000 samples of fixed size 100, 200, 300, and 3,000

the best specimen. Then, we produced 10,000 samples of fixed size 200 and derived new confidence intervals for these specimens. Figure 5 shows that for this sample size, specimens 2 and 10 have also overlapping confidence intervals. Successively building confidence intervals for samples of fixed size 300 and 500, we found that in the latter case these confidence intervals do not overlap. Therefore, for sample size 500 screening rule should classify the specimen 10 as a better one than specimen 2.

However, not always increasing of sample size results in separation of confidence intervals for two specimens.

Figure 6 shows that specimens 1 and 21 have overlapping confidence intervals for sample sizes 100, 200, 300, and 3,000. In this case, the screening procedure cannot select the best specimen. Therefore, both these specimens should be classified as promising steel samples.

This section demonstrated that the approach to CVaR estimation based on distribution built with percentile regression allows calculating confidence interval in addition to point estimate for CVaR. Utilization of confidence intervals in the screening procedure enables to reduce screening errors related with uncertainty in

point estimates of CVaR. For this purpose, we should compare confidence intervals of two specimens instead of their point estimates, as described in Sect. 4. If these confidence intervals do not overlap, these specimens are significantly different. In this case, the specimen, corresponding to the confidence interval with lower values, should be disqualified. However, if these specimens have overlapping confidence intervals, the screening procedure cannot select the best specimen. In this case, we should increase the sample size and repeat the procedure of estimation of confidence interval.

6 Summary and Conclusions

We presented new statistical approaches for modeling mechanical properties and screening specimens in development of new materials with highly variable properties. Paper [1] suggested to use quantile regression for screening specimens. However, quantile regression does not take into account tails of distributions. In contrast to quantile, CVaR is an average of observations exceeding quantile. Therefore, CVaR which takes into account tails may be preferable to quantile for screening specimens.

We investigated two approaches for CVaR estimation. The first approach, based on a Mixed-Quantile Quadrangle, uses experimental data for direct estimating 20 %-CVaR. In particular, we considered a Mixed-Quantile regression for $\ln(CVN)$ with the Rockafellar error function. The second approach estimates 20 %-CVaR in two steps. The first step uses quantile regression for generating a large number of samples of CVN values and building distributions of CVN for each specimen. The second step calculates 20 %-CVaR for each specimen from these distributions.

Accuracy of generated CVN distribution was evaluated with the coefficient of multiple determination, $R^1(\alpha)$, which measures the proportion of the 'variability' in the response that is accounted for the fitted α-th quantile surface. We found that for a wide range of probability levels ($0.05 \leq \alpha \leq 0.832$) the corresponding quantile regression functions capture more than 25 % of the variability in the response (that is accounted for by the fitted α-th quantile surface). The most accurate portion of the generated CVN distribution corresponds to the range of probability levels $0.351 \leq \alpha \leq 0.533$. The quantile regression functions corresponding to this range of α account for 44–48.71 % of the variability in the response.

Discrepancy between estimates of CVaR based on Rockafellar error and on distribution from the percentile regression was measured by the MAE. We found that two approaches to CVaR estimation provide similar estimates of $\mathrm{CVaR}_{0.2}(\ln(\mathrm{CVN}))$ for all specimens. The numerical experiments demonstrated that the average discrepancy between these estimates does not exceed 4.55 %.

We compared performance of these two approaches for CVaR estimation for screening specimens with respect to CVN. The dataset included 33 pairs of specimens. Both approaches resulted in the same ranking for 20 pairs. However,

results of screening were different for 13 pairs. Although the average discrepancy between estimates of $\mathrm{CVaR}_{0.2}(\ln(\mathrm{CVN}))$ with these two approaches does not exceed 4.55 % (see results in the Table 2), discrepancy in screening exceeds 39 %. Since in the second approach we considered 10,000 sampled CVN values for each specimen, while in the first approach we used only 3 values, the second approach is more accurate.

We also analyzed probabilistic models of CVN for different specimens using the following rules of specimens screening:

1. Rule suggested in [3], based on the average and the 20th VaR of the CVN distribution;
2. Rule suggested in [1], based only on the 20th VaR of the CVN distribution;
3. Rule based only on the 20th CVaR of the CVN distribution.

Results of screening by using Rules 1, 2, and 3 sometimes contradicted each other. We think that the Rule 3 which is based on the CVaR of the CVN distribution is the more appropriate, and it should be used for screening process.

CVaR estimation based on distribution (built with percentile regression) allows for calculating confidence interval, in addition to point estimate of CVaR. The confidence intervals in the screening procedure reduce screening errors coming from the uncertainty in estimates of CVaR. We compared confidence intervals of two specimens, instead of their point estimates as described in Sect. 4. If confidence intervals do not overlap, the specimens are significantly different. However, for specimens with overlapping confidence intervals the screening procedure cannot select the best specimen. In this case, we can increase the sample size and repeat the procedure of estimation of confidence interval.

Approaches proposed in this paper could identify promising compositions of materials and processing parameters for developing better steels. This may reduce cost of experimental programs, shifting resources to cost-effective computer modeling techniques.

References

1. Golodnikov, A., Macheret, Y., Trindade, A., Uryasev, S., Zrazhevsky, G.: Statistical Modeling of Composition and Processing Parameters for Alloy Development. Model. Simul. Mater. Sci. Eng. **13**, 633–644 (2005)
2. McClintock, F.A., Argon A.S.: Mechanical Behavior of Materials. Addison-Wesley, Reading, MA (1966)
3. Goldren, A.P., Cox, T.B.: AMAX Report, CPR-2, AMAX Materials Research Center, Ann Arbor, MI (1986)
4. Corwin, W.R., Houghland, A.M.: Effect of Specimen Size and Material Condition on the Charpy Impact Properties of 9Cr-1Mo-V-Nb Steel. In: Corwin, W.R., Lucas, G.E. (eds.) The Use of Small-Scale Specimens for Testing Irradiated Material. ASTM STP 888, Philadelphia, PA, pp. 325–338 (1986)

5. Lucon, E., et al.: Characterizing Material Properties by the Use of Full-size and Sub-size Charpy Tests. In: Siewert, T.A., Manahan, M.P. (Sr. eds.) Pendulum Impact Testing: A Century of Progress. ASTM STP 1380, pp. 146–163. American Society for Testing and Materials, West Conshohocken, PA (1999)
6. Todinov, M.T.: Uncertainty and Risk Associated with the Charpy Impact Energy of Multi-run Welds. Nucl. Eng. Des. **231**, 27–38 (2004)
7. Koenker, R., Bassett, G.: Regression Quantiles. Econometrica **46**, 33–50 (1978)
8. Rockafellar, R.T., Uryasev, S.: The Fundamental Risk Quadrangle in Risk Management, Optimization, and Statistical Estimation. Surv. Oper. Res. Manag. Sci. **18**, 33–53 (2013)
9. Rockafellar, R.T., Uryasev, S.: Optimization of Conditional Value-at-risk. J. Risk **2**, 21–42 (2000)
10. Rockafellar, R.T., Uryasev, S.: Conditional Value-at-risk for General Loss Distributions. J. Bank. Financ. **26**, 1443–1471 (2002)
11. Koenker, R., Machado, J.A.F.: Goodness of Fit and Related Inference Processes for Quantile Regression. J. Am. Stat. Assoc. **94**, 1296–1310 (1999)
12. Taylor, J.W., Bunn, D.W.: Combining Forecast Quantiles Using Quantile Regression: Investigating the Derived Weights, Estimator Bias and Imposing Constraints. J. Appl. Stat. **25**, 193–206 (1998)
13. Bassett, G., Koenker, R.: An Empirical Quantile Function for Linear Models with iid Errors. J. Am. Stat. Assoc. **77**, 407–415 (1982)
14. Koenker, R., d'Orey, V.: Computing Regression Quantiles. Appl. Stat. **43**, 410–414 (1994)